Thesis/
Reports
Taylor,
W.R.

WESTFORNET
MONTHLY ALERT
LOAN COPY "1"
Edition DEC 1984
Item No. 128
File ~~SD~~ ~~538~~ ~~MM0~~ ~~DC 125~~

SIMULATION OF HELISTAT LOAD ASSEMBLY SYSTEMS

128

LOAN COPY "1"

FINAL REPORT

SIMULATION OF HELISTAT LOAD ASSEMBLY SYSTEMS

WILLIAM R. TAYLOR

WESTFORNET
MONTHLY ALERT
Edition DEC 1984
Item No. 128
File SD 538 Mlo no.125

COOPERATIVE RESEARCH PROJECT
Supp. No. INT-81-06-CA

USDA FOREST SERVICE
Intermountain Forest & Range Experiment Station
Ogden, Utah

COLLEGE OF ENGINEERING
Montana State University
Bozeman, Montana

Project No. 3251
Study No. 3251-101,201-X1
David F. Gibson, Principal Investigator

1983

SIMULATION OF HELISTAT LOAD ASSEMBLY SYSTEMS

INTRODUCTION

Recent concerns regarding the environment and conservation in the United States have prompted some governmental agencies to reassess their policies regarding the utilization of natural resources. As a result, the USDA Forest Service, in conjunction with the U.S. Navy, has embarked upon an experimental test program to evaluate the possibility of using a helium-filled balloon with four helicopters attached for harvesting timber in remote, roadless forests. The blimp, which holds one million cubic feet of helium, is attached to the helicopters by an understructure of aluminum tubing. Behind each helicopter is a three-blade propeller to provide propulsion. The entire airship is called a Helistat. With the 1,525 horsepower helicopter engines, it is supposed to be capable of speeds of 80 miles an hour and lifting capacity of 24 tons of timber. Since the Forest Service estimates that about 5 billion board feet a year of diseased and fire-damaged timber cannot be reached via current technology, the development of such a ship could result in substantial rewards. In addition, much of the timber is considered quite small and is located on steep, remote terrain.

The development of a technology capable of harvesting timber under these conditions could result in better utilization of the nation's timber lands.

The ship is currently under construction from war surplus equipment and is scheduled to be tested at several diverse sites in the U.S. Should the concept prove feasible, production models will be designed and built. The machines will be expensive and their economic justification will require that their utilization and productivity be extremely high. Idle time and delays will have to be minimized.

The Intermountain Forest and Range Experiment Station of the Forest Service and Montana State University began a cooperative research project in July 1981 to simulate various phases of the Helistat logging tests to be conducted in the Gallatin and Clearwater National Forests. The simulation model was to primarily address the problem of how to preassemble bunches or loads for removal via the Helistat. However, in order to access overall costs and productivity all phases of the harvesting system were simulated. The results of this modeling research are documented in this report.

PROBLEM STATEMENT

Since the productivity and utilization of expensive equipment must be high in order to economically justify its use, it was required that the simulation model be capable of assessing various methods of prebunching trees into unit loads for later export from the forest. The possibility of "hot logging" was

infeasible due to necessity for assembling up to 20-24 tons of trees within a round trip turn time of 8-10 minutes. Thus, one of the primary tasks was to design the model with sufficient flexibility to be able to simulate a variety of methods and equipment under a diverse set of field conditions. This flexibility required that the user be permitted to input his own cost and productivity data in a number of cases. This kind of system not only allows one to determine the costs and productivity of a particular situation, but also facilitates sensitivity analysis highlighting the impact of various equipment or methods changes. The model also allows the user to investigate various mixes of labor skills as well as the impact of downtime and set-up time.

HELISTAT SYSTEMS MODEL

The Helistat Systems Model is illustrated schematically in Figure 1. There are four primary activities involved in harvesting timber: felling, bunching, yarding, and landing. Also depicted are the inputs and outputs for these primary activities, each of which are dependent upon those activities which chronologically precede it. Thus, it is necessary to consider how trees are felled when determining how best to place them in bunches or loads. Likewise, yarding and loading methods are greatly influenced by felling and bunching.

The arrows indicate the flow of logic or control as well as the flow of information based upon data calculated within each

HELISTAT SYSTEMS MODEL

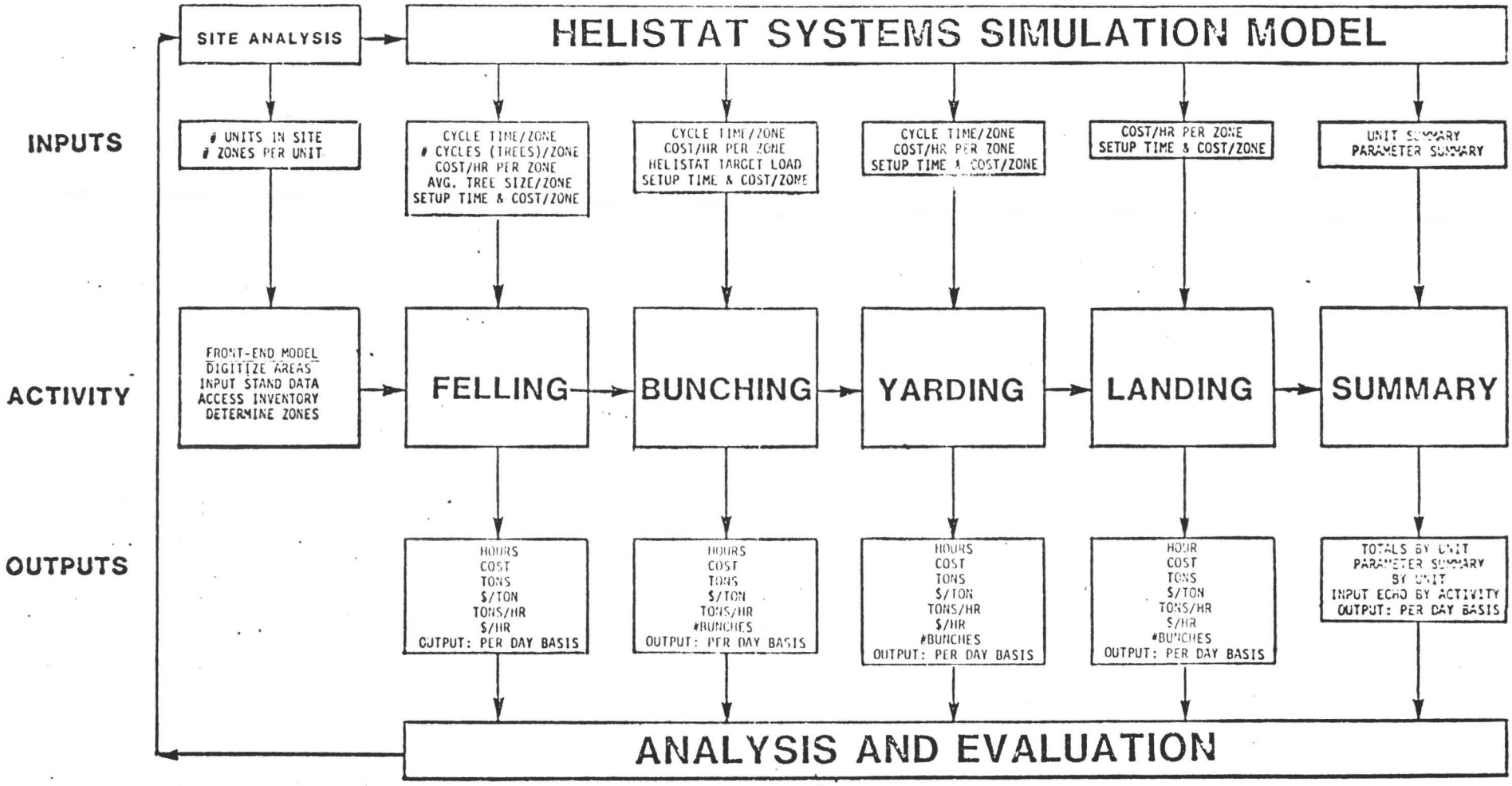


Figure 1. The Helistat Systems Model

activity phase. For example, once the number and size of trees are determined within the felling activity, this information becomes an input to the bunching activity, etc. Each activity phase interactively prompts the user for input data relative to the particular equipment and method being utilized. This along with information from "up stream" activities provides the input needed at each activity phase.

The system provides the user with the option to examine cost and productivity data immediately following the simulation of each activity. The time required to fell or bunch an area is also calculated on a per day basis. Following the bunching, yarding, and loading phases, the number of bunches (loads) required are also output.

A variety of summary reports are available at the conclusion of the simulation which allow the user to assess overall system costs and productivity. For example, one can examine the overall costs associated with only the productive time, or costs related to both productive and set-up costs. Tables on cost per ton, dollars per hour, etc. are available upon request.

Models of any kind are only as good as the assumptions and realism built into them. In order for a simulation model to accurately reflect realism, it must begin with a proper definition of the problem. In this case that means determining the various parameters of the site to be logged as well as determining the various species, sizes, and number of trees on the land. Site parameters would include the size of the area, slope of the terrain, amount of down timber and underbrush,

abrupt slope changes, non-timbered areas, etc. Assembling this kind of information prior to beginning the felling simulation insures that the area to be harvested is well defined. As shown in Figure 1, this initialization process takes place in the "Front-End Model."

THE FRONT-END MODEL

The Front-End Model is menu driven, giving the user a choice of providing input data directly via a CRT and terminal, or by using a digitizing tablet to determine the areas to be logged along with a file containing timber inventory data for that area. The former requires the user to indicate the average cycle time, set-up time, cost per hour, etc. for each area to be logged and circumvents the front-end model. The second method provides a more accurate assessment of the timber actually in each stand, as well as a very accurate calculation of the exact number of acres to be harvested. The latter approach facilitates starting the simulation with "real data." A more detailed discussion of it follows.

Digitizing Areas

Once a contour map of the area to be logged is obtained, each Cutting Unit along with the Timber Stands within them can be identified on the map. Figure 2 illustrates a typical situation. In this example, Cutting Unit 1 is partitioned into two zones indicating steep and gentle slopes while Cutting Unit 2 consists of one zone. Zones 1 and 2 in Cutting Unit 1 are separated by a

section of Non-Commercial Forest (NCF). Stands are identified by numbered areas. The user interactively responds to questions and instructions regarding the digitizing of the Cutting Unit, Zone 1, Stands within Zone 1, etc. The model then automatically calculates the area of each piece of terrain and stores it for later use. Thus, each area is tied to a particular geographical location and has a known size and shape. Additional data such as the elevation of each area could also be acquired and stored during the digitizing process.

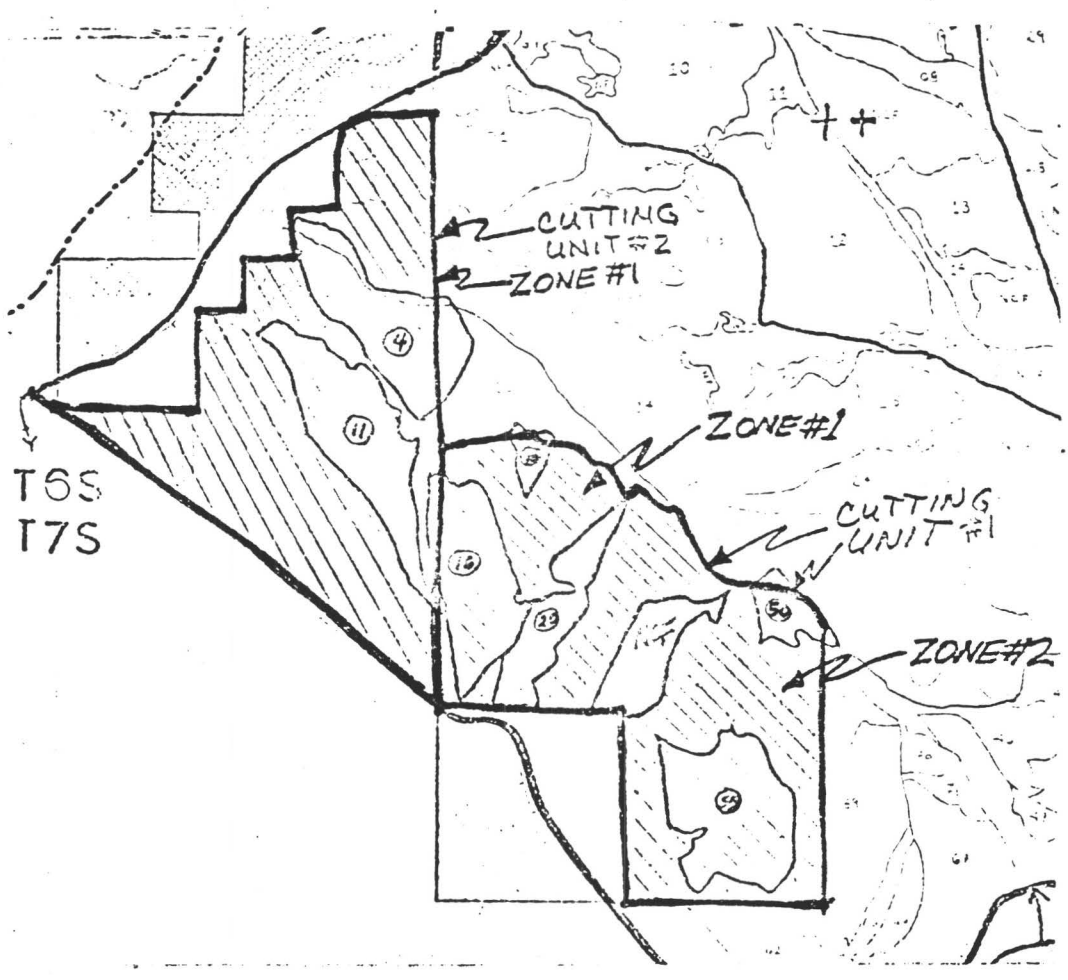


Figure 2 - Typical Gallatin Forest Site - Levenski Drainage

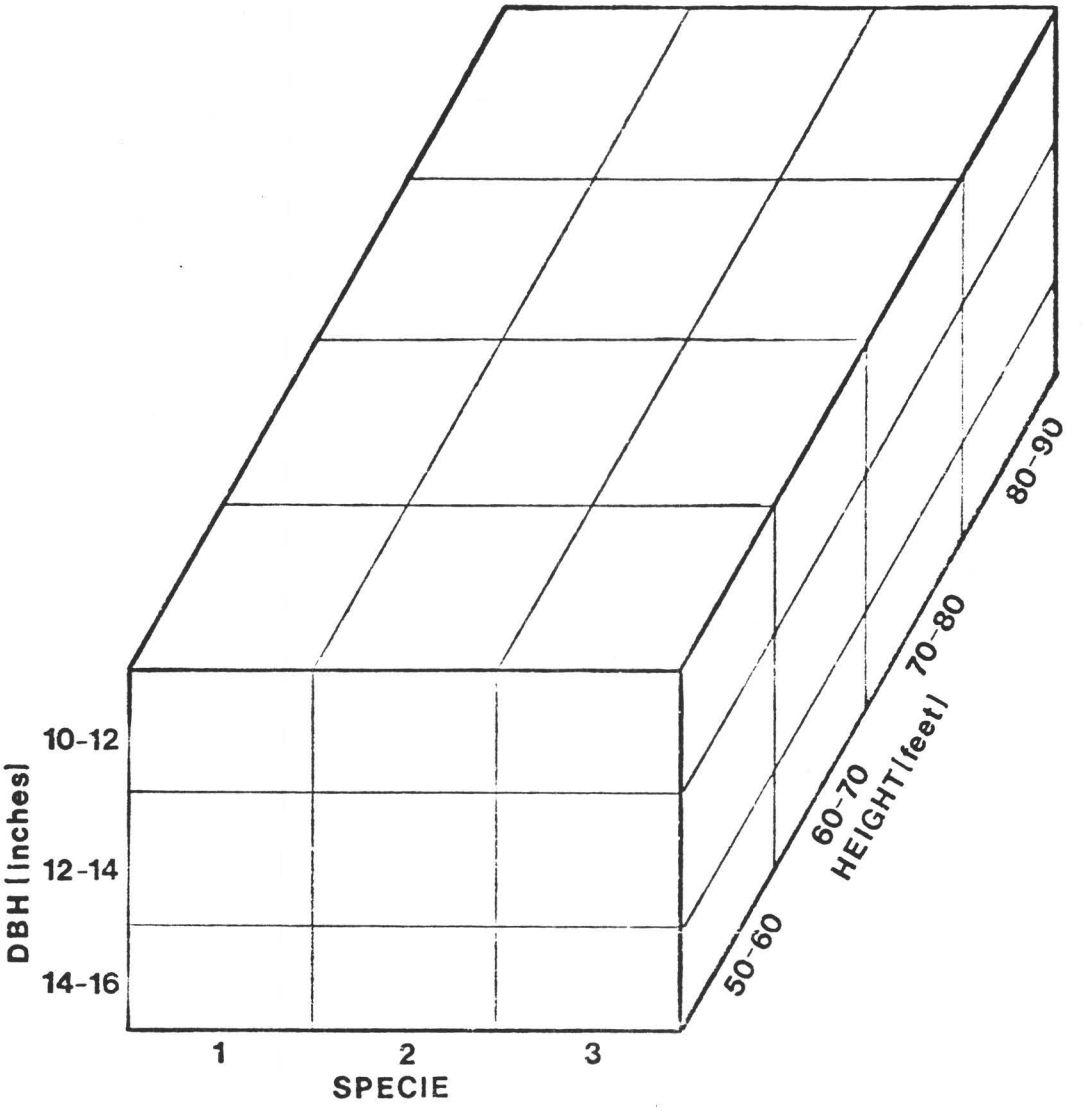
Timber Characteristics

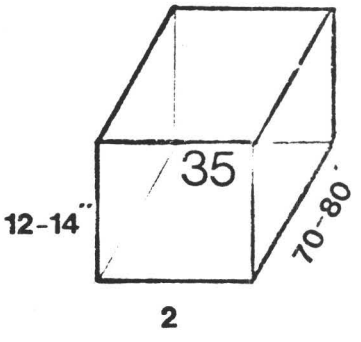
Stands Examinations and Cruise Studies are standard operating procedures prior to offering timber for sale. Among other paramters, these sampling studies determine the species, top and DBH diameters, heights, etc. of a representative sample within each stand. The number of trees in each category are then expressed on a per acre basis. This information along with whether the trees are alive cr dead allows one to accurately calculate the kind and volume of timber in each stand.

For example, suppose that a stand has 10 acres and contains 50 live lodgepole trees per acre which are 6 inches at the top and 10 inches DBH with a height of 60 feet. Then, (10 acres) (50 trees/acre) = 500 trees with these particular characteristics in the stand. The diameters and heights are used to calculate timber volume and weight. When all species and sizes are considered, a very accurate assessment can be made of the inventory in the stand.

In order to have this kind of data available, it was necessary to design a file which would provide distributions on DBH, height, specie, and live (vs) dead illustrated in Figure 3 as the axes of a cube.

Each cell within the cube represents a particular combination of these three parameters and the information contained within each cell reflects the number of trees per acre with those characteristics. Dead timber is identified by preceding the cell entry by a minus sign.





Each cell contains trees per acre.

Figure 3. Timber Characteristics Data File Structure

Once constructed this timber inventory data file is used to calculate values for the following variables which are inputs to the felling activity phase of the model: total volume of the timber, total weight of the trees, average and standard deviation of tree weight, number of trees to be processed, and the number of Helistat loads required to remove the timber.

Since Helistat operations may take place in a variety of locations, it was necessary to structure a unique identification coding scheme for any site in the U.S. Figure 4 illustrates how the system works. As shown in the example, a unique identification number can be specified for each timber stand in the U.S. Once the related timber data is entered into the file (via an interactive program called BSDAT) it can be recalled upon demand for use in the simulation model. When matched up with the area, size, and shape information acquired via digitization a complete assessment of the timber and land are available.

FELLING

Inputs to the Felling activity phase come from one of two sources. The user is given the option of using a digitizing tablet along with the timber data file previously described, or inputting values in response to inquiries directly from a terminal or keyboard. This flexibility allows the user to simulate any kind of felling system by inputting costs and productivity figures for it. This data may be different for each cutting unit, zone, stand, etc.

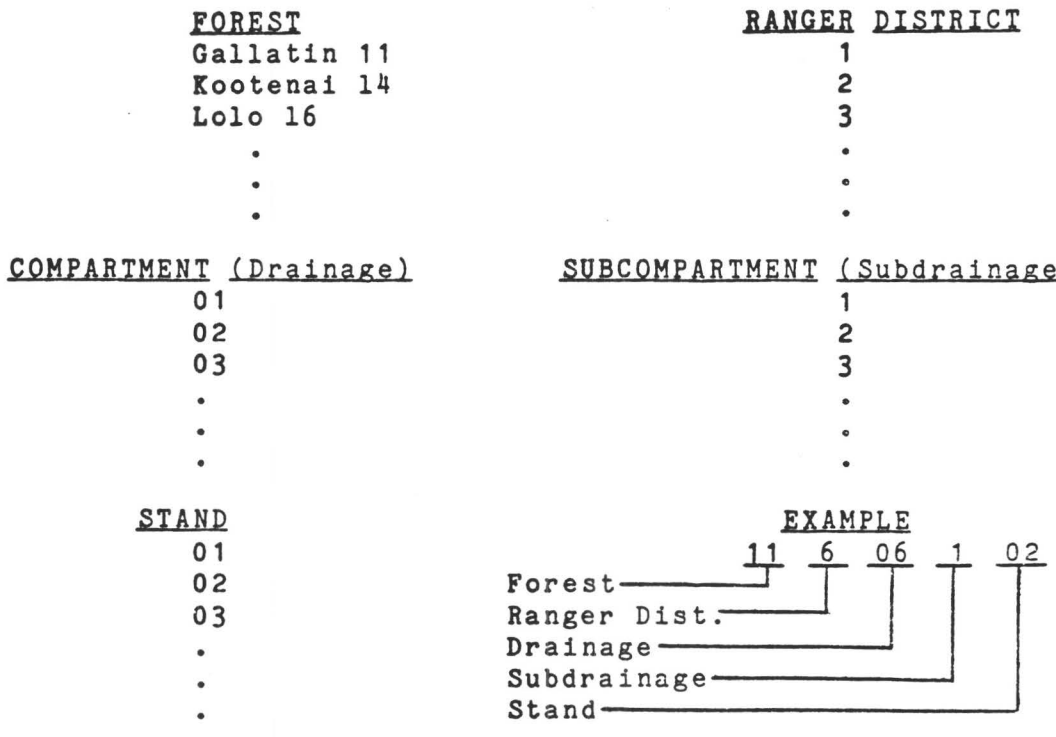


Figure 4 - Site Identification Coding Scheme

After the data is input, the user is provided a "menu" from which to choose an output report on felling. Productivity and cost reports are available on a per hour or per day basis.

BUNCHING

Calculations during the Felling phase result in data that is then passed on to the Bunching operations. In addition, the user is interactively prompted for additional information relating to the bunching equipment to be used.

With the input data in the machine, the menu driven output option is again presented -- this time for Bunching only.

YARDING

Yarding consists of transporting the timber from the forest to a landing. In addition to the information calculated during

the Felling and Bunching operations which is passed downstream, the user again responds to questions regarding the operation of the Helistat in this environment.

The model calculates the number of trips required to yard the timber in each Zone as well as in each Cutting Unit. Costs and productivity data are also available as in previous phases.

LANDING

The activity at the Landing involves processing each load deposited by the Helistat. The timber must be cleared from the landing prior to the arrival of the next load from the forest. Also a continuing operation is the preparation and loading of trees on logging trucks for transportation to a mill. Thus, the landing operations involve rather large material handling equipment. The costs and capabilities of this equipment is supplied by the user in response to questions from the model.

Output reports on the Landing operations are available and similar to those from the other upstream activities.

SUMMARY REPORTS

After the model has simulated production at each activity phase considering their dependent relationships, the overall results are available in a variety of formats.

Unit Summary Report

As the name implies this is a report by Cutting Unit. The harvesting time is divided into its productive and setup components. For each component the hours required, tons of

timber harvested, the overall costs, and the cost per ton are calculated and displayed.

Unit Activity Summary

The Unit Activity Summary provides the user with breakdown of the hours required and the associated cost for each of the activity phases -- Felling, Bunching, Yarding, and Landing. The output for each Cutting Unit is presented separately and will be illustrated later by several comprehensive examples.

Parameter Summaries

The output options also allow the user to examine five different parameters at the conclusion of the simulation. Reports are available on time, both productive and setup hours, by Cutting Unit. This allows the user to see the time and percentage of time spent on each of the activities' phases. Similar reports are available for cost, dollars per ton, tons per hour, and dollars per hour.

These reports assist the user in determining the high cost, low productivity activities in the harvesting system and indicate where management attention may be required.

Echo Checks

Lastly, the model outputs all of the data that was input initially by the user. This provides a validation procedure and helps insure the viability of the model.

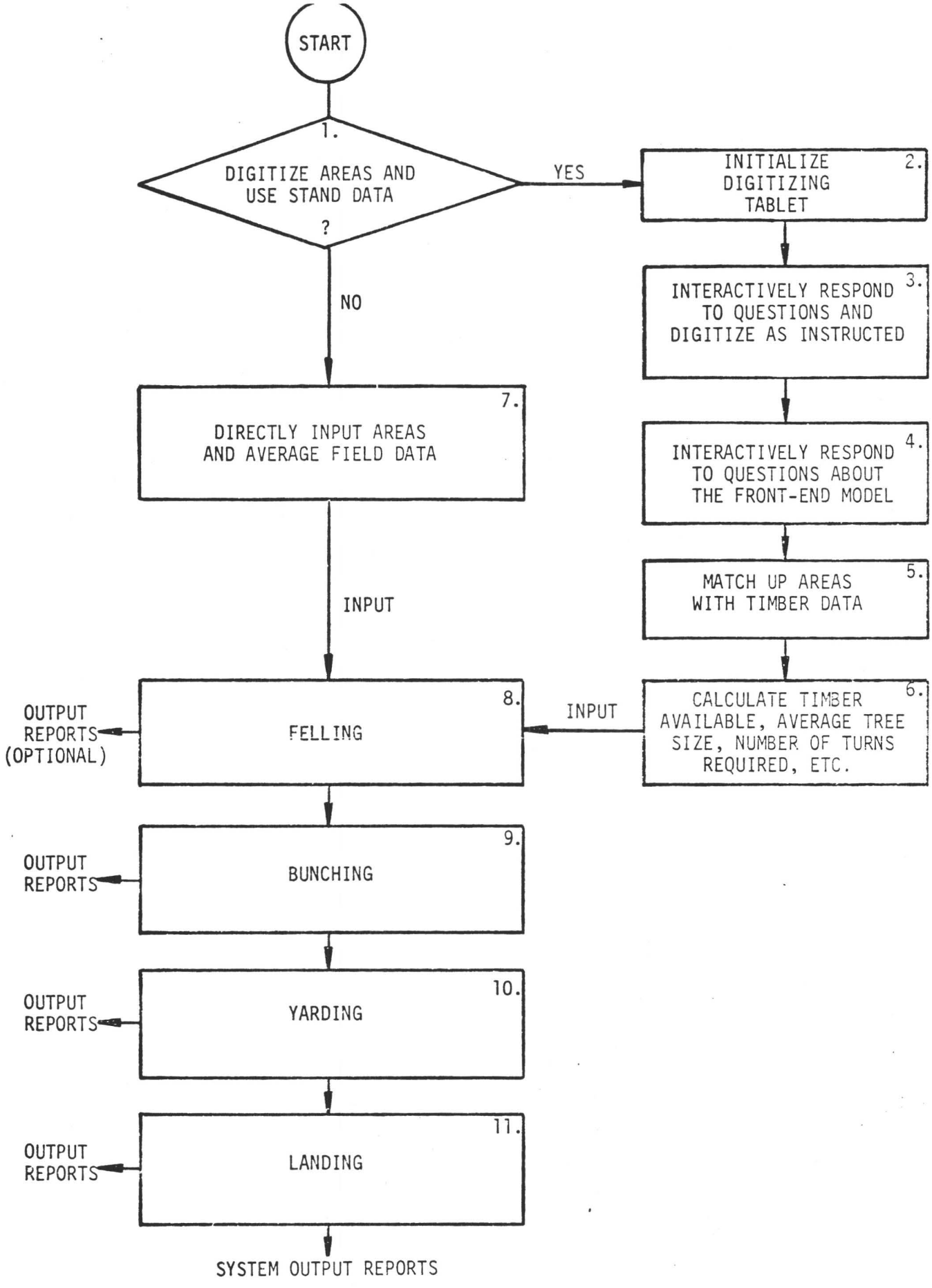


Figure 5 - Systems Model Logic

SYSTEM SUMMARY

Although Figure 1 illustrates the structure of the Helistat Systems Model, the flowchart of Figure 5 better summarizes the program logic. Soon after the program begins, the user is given the option of using his own estimates or average values for areas and characteristics of the timber. This option facilitates use of the model when only "rough" data are available, allowing the user to gain a "feel" for cost and productivity figures based on a number of "what-if" situations. Of course, the estimates may also be quite refined and provide excellent results.

Conversely, the user may have access to digitizing equipment and computers capable of storing a timber data base for use with the Front-End Model. This option allows the matching of each cutting unit, stand, and zone area with quite accurate cruise data. Including good estimates of equipment and labor costs should yield accurate information upon which to base operating decisions.

Regardless of how this initial data is introduced into the model, both approaches then proceed by simulating the Felling, Bunching, Yarding, and Landing activities, providing optional output reports on costs and productivity for each phase.

Probably the best way to understand the overall systems model is to illustrate it with comprehensive examples involving harvesting of the areas included in Figures 2 and 11, which represent typical Gallatin and Clearwater Forest sites, respectively.

GALLATIN FOREST EXAMPLE

The model is initiated by running the main program, HEST4, which immediately outputs the contents of Figure 6. Next the user responds to the statement:

NUMBER OF CUTTING UNITS IN THE SITE=__2*__.

The program then asks the user to choose the method of initiating the simulation as follows:

```
ENTER THE NUMBER CORRESPONDING TO THE DESIRED OPTION
**************************************************************************
 1 • DIGITIZE TO DETERMINE AREAS & USE ACTUAL STAND DATA FOR CALCULATIONS

 2 • INPUT AVERAGES FOR AREAS & FIELD DATA FROM THE TERMINAL

**************************************************************************
```

Digitizing Areas

Prior to beginning the digitizing of each area several files and paramters must be established as follows:

```
TYPE 6 CHARACTER OUTPUT FILE NAME,THEN ::25DEC201::25
Type a title for the data file -> DEC221
What is the X coordinate of lower left corner? 0
What is the Y coordinate of lower left corner? 0
What is the X coordinate of upper right corner? 10560
What is the Y coordinate of upper right corner? 10560
```

The X and Y coordinates correspond to the dimensions of the area being digitized.

Next the overall boundary of the portion to be digitized is established by the cursor as follows:

DIGITIZE THE LOWER LEFT-HAND AND UPPER RIGHT-
HAND POINTS ON THE MAP.

*underlined numbers indicate the user's input for the example.

HELISTAT SYSTEMS SIMULATION MODEL

DEVELOPED BY

COLLEGE OF ENGINEERING

MONTANA STATE UNIVERSITY

BOZEMAN, MONTANA

IN COOPERATION WITH

USDA FOREST SERVICE

INTERMOUNTAIN FOREST & RANGE EXPERIMENT STATION

MISSOULA, MONTANA

1983

PRESS THE RETURN KEY TO CONTINUE

Figure 6 - Title Page for the Model

Now the digitizing of the areas begins with the following statement.

> WE ARE NOW READY TO DIGITIZE ONE AREA AT A TIME. WHEN COMPLETED WITH EACH AREA PRESS THE SPACE THEN THE RETURN KEYS ON THE TEKTRONIX TERMINAL.
>
> DIGITIZE THE BOUNDARY OF CUTTING UNIT 1.

Next the user is given the opportunity to divide the cutting unit into zones which distinguish diverse topographical or timber characteristics.

```
DO YOU WISH TO PARTITION THE CUTTING UNIT INTO
ZONES TO DISTINGUISH DIFFERENT TOPOGRAPHY?
YE=YES, NO=NO
```

A response of YE is followed by this question:

```
HOW MANY ZONES ARE DESIRED?
```

If no partitioning was requested the entire cutting unit is treated as Zone 1. The next instruction concerns the first zone.

```
DIGITIZE THE BOUNDARY OF ZONE 1 OF CUTTING
UNIT 1.
```

Next the stands within each zone are digitized as follows:

```
ENTER THE NUMBER OF STANDS TO BE LOGGED IN
ZONE 1.  3
```

With the number of stands now known the program will ask the user to specify a stand number and then digitize each one as follows:

```
ENTER THE STAND NUMBER OF THE NEXT STAND TO BE
DIGITIZED IN ZONE 1 OF CUTTING UNIT 1.  16
```

```
DIGITIZE THE PORTION OF STAND 16 THAT IS
WITHIN ZONE 1 OF CUTTING UNIT 1.
```

Frequently, only a portion of a stand may be within a zone or cutting unit boundary. Digitizing facilitates an accurate assessment of the exact area to be logged. Figure 7 illustrates the status after digitizing Cutting Unit 1, Zone 1, and Stands 16, 18, and 20 within Zone 1. Note that some stands within Zone 1 are not being harvested.

After each stand to be harvested from Zone 1 has been digitized, the user is interactively instructed to digitize Zone 2 and each stand within it. Additional zones are processed similarly.

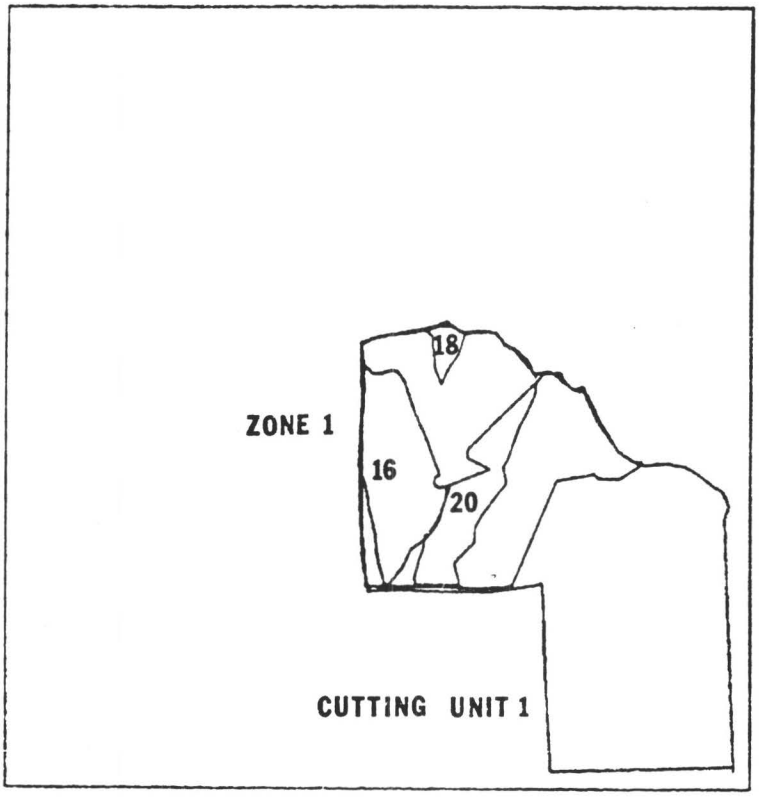


Figure 7 - Digitized Map of Zone 1 and Stands

Once all zones and stands within Cutting Unit 1 have been digitized, the user is instructed as follows:

DIGITIZE THE BOUNDARY OF CUTTING UNIT 2.

The user is given the opportunity to partition the unit into zones and then to digitize each stand within each zone as previously illustrated. When all of the cutting units have been digitized the program calculates the area (acres) of each stand, zone and cutting unit and prints out the information on the completed digitized map as shown in Figure 8.

When the return key is pressed the program transfers to the Front-End Model to determine the characteristics of the timber within each stand.

THE CALCULATED AREAS:

AREA OF CUTTING UNIT	1 -	484.0249
AREA OF ZONE	1 -	231.6435
AREA OF STAND	16 -	54.4048
AREA OF STAND	18 -	5.2925
AREA OF STAND	20 -	34.2596
AREA OF ZONE	2 -	223.7122
AREA OF STAND	50 -	13.3325
AREA OF STAND	55 -	67.2727
AREA OF CUTTING UNIT	2 -	420.5226
AREA OF ZONE	1 -	420.5226
AREA OF STAND	4 -	54.6360
AREA OF STAND	11 -	67.3739

PRESS THE RETURN KEY TO CONTINUE

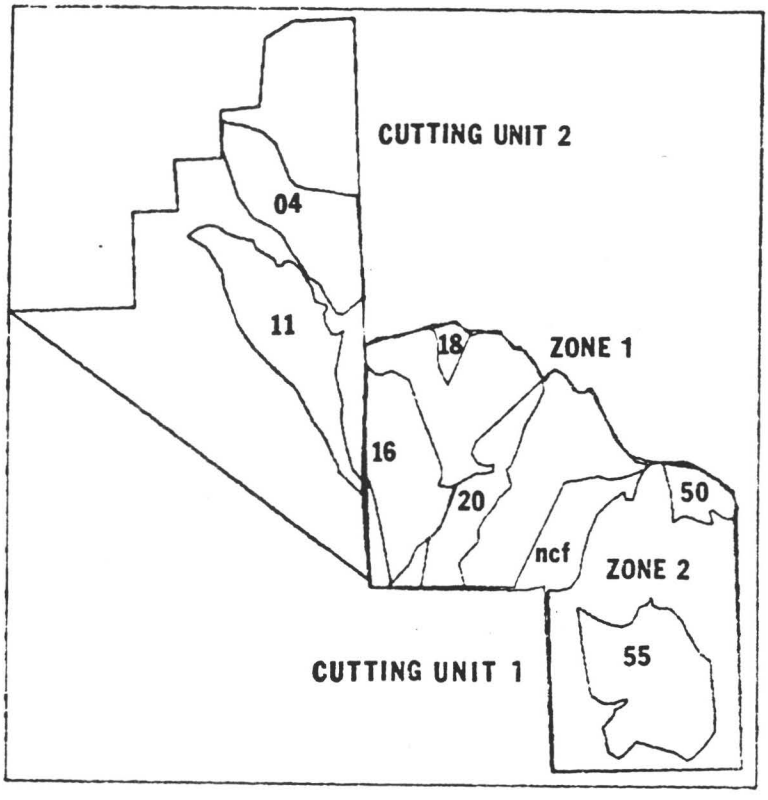


Figure 8 - Complete Digitized Map

Front-End Model

In order for the Front-End Model to work in conjunction with the digitized areas, a data file must have already been built. Each data file has a unique name identifying a particular location in the forest. The key piece of information is the number of trees per acre of each specie, height and DBH class. When multiplied by the number of acres obtained by digitizing, the result is the number of trees of each type and size within

the stand. This data can be used to calculate the tons of timber to be harvested, bunched, and transported from each stand.

A fictitious set of data has been structured for this example problem. An actual set of data would be of identical form except that it would probably contain more entries. The data base is shown in Figure 9. Each set of stand data starts with a "header" record that begins with a # sign and indicates the exact location of the stand, the stand number, approximate number of acres, and the number of data entries to follow. Each data entry indicates the specie, height class, diameter class, and number of trees per acre. Appendix 1 provides additional information on data bases.

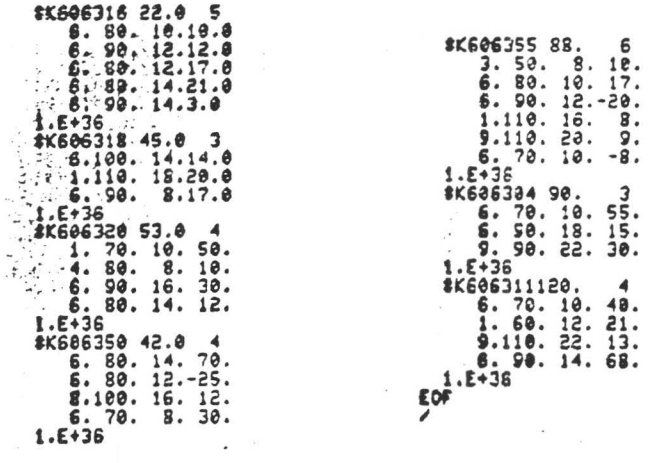


Figure 9 - Typical Data Base for Example Problem

The program next asks the user to precisely identify the location to be harvested:

```
THE FOLLOWING CONCERNS CUTTING UNIT 1.
ENTER THE NUMBER OF DIGITS INDICATED.
FOREST NUMBER = 11
BLOCK OR RANGER DISTRICT NUMBER = 6
COMPARTMENT OR MAIN DRAINAGE NUMBER = 06
SUBCOMPARTMENT OR SUBDRAINAGE NUMBER = 03

ENTER THE NUMBER OF STANDS IN ZONE 1   3
ENTER A STAND NUMBER OF THE FIRST(NEXT) STAND
IN ZONE 1 OF CUTTING 1   16
```

The program then searches the timber data base (Figure 9) for a header file that matches these numbers, retrieves the data relative to Stand 16 and writes it out to a file identifying it as data for Zone 1. This process is repeated until each stand in Zone 1 is located and retrieved. Then a similar routine processes Zone 2 and its stands. Cutting Unit 2 would then be handled in a similar manner. Figure 10 shows how the files representing each stand (as shown in Figure 9) have been sorted into separate zone files for further processing.

At this point in the example problem, the Front-End Model has prepared files which can be used to calculate the average tree size per zone and the total weight (tons) of trees to be processed per zone. On Figure 5, the program has now progressed through step 6. However, the user is again given the option of using this data in further operations or providing independent estimates of the same.

```
DO YOU WISH TO INPUT THE FELLING CYCLE TIMES OR USE THE DEFAULT VALUES FROM EMPIRICAL DATA ?
1 = INPUT MY VALUES        2 = USE DEFAULT VALUES
?
```

Responding with a "2" causes the program to use the timber characteristic data as it moves on to the felling operation.

```
$K6063    UNIT 1    ZONE 1
      6.00      84.95      10.00      10.00
      6.00      94.95      12.00      12.00
      6.00      84.95      12.00      17.00
      6.00      84.95      14.00      21.00
      6.00      94.95      14.00       3.00
      6.00     104.95      14.00      14.00
      1.00     114.95      18.00      20.00
      6.00      94.95       8.00      17.00
      1.00      74.95      10.00      50.00
      4.00      84.95       8.00      10.00
      6.00      94.95      16.00      30.00
      6.00      84.95      14.00      12.00
THERE ARE12 ENTRIES IN THE DATA FILE FOR ZONE  1
$K6063    UNIT 1    ZONE 2
      6.00      84.95      14.00      70.00
      6.00      84.95      12.00     -25.00
      8.00     104.95      16.00      12.00
      6.00      74.95       8.00      30.00
      3.00      54.95       8.00      10.00
      6.00      84.95      10.00      17.00
      6.00      94.95      12.00     -20.00
      1.00     114.95      16.00       8.00
      9.00     114.95      20.00       9.00
      6.00      74.95      10.00      -8.00
THERE ARE10 ENTRIES IN THE DATA FILE FOR ZONE  2

$K6063    UNIT 2    ZONE 1
      6.00      74.95      10.00      55.00
      6.00      94.95      18.00      15.00
      9.00      94.95      22.00      30.00
      6.00      74.95      10.00      40.00
      4.00      64.95      12.00      21.00
      9.00     114.95      22.00      13.00
      6.00      94.95      14.00      68.00
THERE ARE 7 ENTRIES IN THE DATA FILE FOR ZONE  1
```

Figure 10 - Sorted Timber Data Files for Each Zone in the Example

However, prior to examining the felling activities, suppose that the user had desired not to use the digitizing equipment and the Front-End Model. As shown in Figure 5 it is possible to arrive at operation number 8, felling, by directly inputting areas and other data.

Input Via Terminal

Consider how this example problem might be solved by responding with a "1" to the following request. This option (Step 7 in Figure 5) requires the user to interactively respond to the following questions:

```
ENTER THE TOTAL  NUMBER OF  ACRES IN CUTTING
UNIT 1    484.02
DO YOU DESIRE TO PARTITION CUTTING UNIT 1 INTO
ZONES TO DISTINGUISH DIFFERENT TOPOGRAPHY:
YE=YES, NO=NO  YES

HOW MANY ZONES ARE DESIRED?  2
ENTER THE TOTAL ACRES IN ZONE 1 OF CUTTING
UNIT 1    231.64

ENTER  THE NUMBER  OF STANDS  TO BE  LOGGED IN
ZONE 1 OF CUTTING UNIT 1     3
ENTER  STAND NUMBER FOLLOWED BY THE STAND AREA
THAT  LIES  WITHIN ZONE 1.  SEPARATE DATA WITH
COMMA  16,50,40
ENTER STAND NUMBER FOLLOWED BY  THE STAND AREA
THAT LIES WITHIN ZONE 1.  20,34,25

ENTER  THE  TOTAL  ACRES  IN ZONE 2 OF CUTTING
UNIT 1    223.71
```

The program continues to query the user for similar information regarding each zone, stand, and cutting unit until all data has been entered. When this task is completed the program is at step 8, felling, in Figure 5. Thus, one can arrive at this position in the program either via digitizing and the Front-End Model, or by inputting data directly from the terminal. Regardless, the next step is to investigate how this data will impact the felling operation.

Felling

The user is asked to provide additional information regarding the felling operation itself. If the first question is answered with a "2" the felling cycle time used in the model is one calculated from empirical data. Other times may be substituted as required. The following information would be requested of the user:

```
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 1•30
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 1 •1.1
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 1 •20
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 2 OF UNIT 1•40
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 2 OF UNIT 1 •.8
SYSTEM SET-UP COST ($/HR) FOR ZONE 2 OF UNIT 1 •35
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2 •7.5
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 2•65
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 2 •.5
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 2 •35
TOTAL WEIGHT(TONS) OF TREES IN ZONE 1 •       4101.3

AVERAGE TREE SIZE/LBS) IN ZONE 1 •       1140.5
 PRESS RETURN KEY TO CONTINUE
 TOTAL WEIGHT(TONS) OF TREES IN ZONE 2 •       3893.1

AVERAGE TREE SIZE(LBS) IN ZONE 2 •       3481.7
 PRESS RETURN KEY TO CONTINUE
 TOTAL WEIGHT(TONS) OF TREES IN ZONE 1 •       13521.9

AVERAGE TREE SIZE(LBS) IN ZONE 1 •       1799.2
 PRESS RETURN KEY TO CONTINUE
```

Use of the default values means that the total tonnage of trees per zone and the average weight of trees per zone are calculated and printed directly from the sorted timber data files previously established. Indicating that one desires to input his own felling cycle times, etc., would cause the program to respond with the following questions.

```
FELLING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 1•3.5
NUMBER OF CYCLES FOR ZONE 1 OF UNIT 1•300
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 1•30
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 1 •1.1
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 1 •20
AVERAGE TREE SIZE(POUNDS) FOR ZONE 1 OF UNIT 1•890


FELLING CYCLE TIME(MIN)FOR ZONE 2 OF UNIT 1•3.8
NUMBER OF CYCLES FOR ZONE 2 OF UNIT 1•300
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 2 OF UNIT 1•40
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 2 OF UNIT 1 •.8
SYSTEM SET-UP COST ($/HR) FOR ZONE 2 OF UNIT 1 •35
AVERAGE TREE SIZE(POUNDS) FOR ZONE 2 OF UNIT 1•1000


FELLING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 2•4
NUMBER OF CYCLES FOR ZONE 1 OF UNIT 2•300
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2 •87.5
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 2•65
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 2 •.5
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 2 •35
AVERAGE TREE SIZE(POUNDS) FOR ZONE 1 OF UNIT 2•900
```

Notice that the user is required to specify the number of felling cycles (trees), the cycle time per tree, and the average tree size within each zone. This information was all calculated from the data when the other option was specified.

Calculations are then made concerning the felling activity with the pertinent data passed on to the bunching, yarding, and landing sections of the model.

Next the user is presented a menu as follows for selecting the kind of output data desired.

```
ENTER NUMBER FOR OUTPUT DESIRED

0=NONE
1=PRODUCTIVITY AND COSTS BASED ON PER HOUR DATA
2=PRODUCTIVITY AND COSTS BASED ON PER DAY BASIS
```

OUTPUTS (Felling)

Selection of "1" yields the following report:

FELLING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR
ZONE(1)	427.92	4101.30	12837.54	3.13	9.58	30.00
ZONE(2)	133.06	3893.05	5322.38	1.37	29.26	40.00
TOTALS	560.98	7994.35	18159.92	2.27	14.25	32.37

FELLING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR
ZONE(1)	894.33	13521.86	58131.22	4.30	15.12	65.00
TOTALS	894.33	13521.86	58131.22	4.30	15.12	65.00

PRESS RETURN KEY TO CONTINUE

Choosing a "2" results in a report based on a per day basis as follows. Note that set-up times and costs are included in this report.

FELLING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.90	1.10	62.02	12837.54	1386.00	14223.54	4101.30
ZONE(2)	7.20	.80	18.48	5322.38	532.00	5854.38	3893.05
TOTAL			80.50	18159.92	1918.00	20077.92	7994.35

FELLING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	7.00	.50	127.76	58131.22	2240.00	60371.22	13521.86
TOTAL			127.76	58131.22	2240.00	60371.22	13521.86

PRESS RETURN KEY TO CONTINUE

Each time the return key is pressed the "menu" is again presented until a "0" is entered telling the computer to progress on to the next position (bunching) of the model.

Bunching

The user is next asked to input data relating to how long it takes to bunch each tree within each zone, as well as some cost data. One additional question concerns the size of the bunches (the Helistat target load). This is equivalent to the amount to be lifted and transported by the airship per trip. A different target load may be specified for each zone in order to account for variations in altitude, terrain features, etc. The questions are as follows:

```
INPUT THE BUNCHING DATA


BUNCH CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 1=4
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1=8
SYSTEM BUNCHING COST/HR FOR ZONE 1 OF UNIT 1=45
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1=.5
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 1=35
HELISTAT TARGET LOAD IN POUNDS=9000


BUNCH CYCLE TIME(MIN)FOR ZONE 2 OF UNIT 1=3.6
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1=7.5
SYSTEM BUNCHING COST/HR FOR ZONE 2 OF UNIT 1=45
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1=.8
THE SYSTEM SETUP COST ($/HR) FOR ZONE 2 OF UNIT 1=40
HELISTAT TARGET LOAD IN POUNDS=8500


BUNCH CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 2=4.2
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2=8
SYSTEM BUNCHING COST/HR FOR ZONE 1 OF UNIT 2=50
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2=.75
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 2=45
HELISTAT TARGET LOAD IN POUNDS=10000
```

Actually, the bunching algorithm would work well with yarding systems other than the Helistat, such as with helicopter or cable systems. Output reports similar to those shown for felling are available via a menu.

BUNCHING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	$ BUNCHES
ZONE(1)	479.46	4101.30	21575.70	5.26	8.55	45.00	912.00
ZONE(2)	134.18	3893.05	6038.00	1.55	29.01	45.00	917.00
TOTALS	613.64	7994.35	27613.70	3.45	13.03	45.00	1829.00

BUNCHING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	$ BUNCHES
ZONE(1)	1052.15	13521.86	52607.44	3.89	12.85	50.00	2705.00
TOTALS	1052.15	13521.86	52607.44	3.89	12.85	50.00	2705.00

PRESS RETURN KEY TO CONTINUE

BUNCHING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	7.50	.50	63.93	21575.70	1120.00	22695.70	4101.30
ZONE(2)	6.70	.80	20.03	6038.00	672.00	6710.00	3893.05
TOTAL			83.95	27613.70	1792.00	29405.70	7994.35

BUNCHING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	7.25	.75	145.12	$2607.44	4927.50	57534.94	13521.86
TOTAL			145.12	52607.44	4927.50	57534.94	13521.86

PRESS RETURN KEY TO CONTINUE

Yarding

The yarding operation involves the Helistat transporting a load from the bunching area to a landing or a mill. The yarding cycle time is a function of many variables: distance to loading, route, temperature, wind conditions, elevation, load, etc. By allowing the user to specify this parameter a great deal of flexibility is attained which facilitates sensitivity analysis. Some cost data is also requested.

```
INPUT YARDING DATA


YARDING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 1•6.4
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1•7
SYSTEM YARDING COST/HR FOR ZONE 1 OF UNIT 1•100
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1•.9
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 1•150


YARDING CYCLE TIME(MIN)FOR ZONE 2 OF UNIT 1•5.3
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1•7.5
SYSTEM YARDING COST/HR FOR ZONE 2 OF UNIT 1•125
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1•1
THE SYSTEM SETUP COST ($/HR) FOR ZONE 2 OF UNIT 1•110


YARDING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 2•3.2
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2•7.5
SYSTEM YARDING COST/HR FOR ZONE 1 OF UNIT 2•145
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2•1.10
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 2•100
```

Optional output reports are available concerning the costs and productivity of the yarding operation.

YARDING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	97.28	4101.30	9728.00	2.37	42.16	100.00	912.00
ZONE(2)	81.00	3893.05	10125.21	2.60	48.06	125.00	917.00
TOTALS	178.28	7994.35	19853.21	2.48	44.84	111.36	1829.00

YARDING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	144.27	13521.86	20918.67	1.55	93.73	145.00	2705.00
TOTALS	144.27	13521.86	20918.67	1.55	93.73	145.00	2705.00

PRESS RETURN KEY TO CONTINUE

YARDING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.10	.00	15.95	9728.00	2160.00	11888.00	4101.30
ZONE(2)	6.50	1.00	12.46	10125.21	1430.00	11555.21	3893.05
TOTAL			28.41	19853.21	3590.00	23443.21	7994.35

YARDING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.40	1.10	22.54	20918.67	2530.00	23448.67	13521.86
TOTAL			22.54	20918.67	2530.00	23448.67	13521.86

PRESS RETURN KEY TO CONTINUE

Landing

At the landing the trees delivered by the Helistat are sorted and either stacked or loaded onto vehicles for

transportation to a mill. If the Helistat flies to a mill, only the stacking and sorting operations would be required. The input and optional output for the landing operations are as follows:

```
INPUT THE LANDING DATA


SYSTEM LANDING COST/HR FOR ZONE 1 OF UNIT 1=80
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1=.5
THE SYSTEM SETUP COST PER HOUR FOR ZONE 1 OF UNIT 1=50


SYSTEM LANDING COST/HR FOR ZONE 2 OF UNIT 1=90
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1=.75
THE SYSTEM SETUP COST PER HOUR FOR ZONE 2 OF UNIT 1=55


SYSTEM LANDING COST/HR FOR ZONE 1 OF UNIT 2=110
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2=1
THE SYSTEM SETUP COST PER HOUR FOR ZONE 1 OF UNIT 2=35
```

LANDING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	97.28	4101.30	7782.40	1.90	42.16	80.00	912.00
ZONE(2)	81.00	3893.05	7290.15	1.87	48.06	90.00	917.00
TOTALS	178.28	7994.35	15072.55	1.89	44.84	84.54	1829.00

LANDING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	144.27	13521.86	15869.33	1.17	93.73	110.00	2705.00
TOTALS	144.27	13521.86	15869.33	1.17	93.73	110.00	2705.00

RESS RETURN TO CONTINUE

LANDING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.10	.50	15.95	7782.40	400.00	8182.40	4101.30
ZONE(2)	6.50	.75	12.46	7290.15	536.25	7826.40	3893.05
TOTAL			28.41	15072.55	936.25	16008.80	7994.35

LANDING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.40	1.00	22.54	15869.33	805.00	16674.33	13521.86
TOTAL			22.54	15869.33	805.00	16674.33	13521.86

RESS RETURN TO CONTINUE

After progressing through felling, bunching, yarding and landing, the user is given an option of reviewing two types of summaries. Each is illustrated below:

UNIT SUMMARY REPORT

		HOURS	TONS	COST	$/TON
UNIT(1)	PRODUCTIVE	1531.18	7994.35	80699.39	10.09
	SETUP	178.45		8236.25	
	TOTAL	1709.63	7994.35	88935.64	11.12
UNIT(2)	PRODUCTIVE	2235.02	13521.86	147526.7	10.91
	SETUP	221.80		10502.50	
	TOTAL	2456.82	13521.86	158029.2	11.69
SITE TOTALS	PRODUCTIVE	3766.19	21516.21	228226.0	10.61
	SETUP	400.25		18738.75	
	TOTAL	4166.44	21516.21	246964.8	11.48

PRESS RETURN KEY TO CONTINUE

UNIT ACTIVITY SUMMARY

(PRODUCTIVE HOURS & COSTS)

UNIT#	AREA (ACRES)	(TONS) VOLUME	FELLING HOURS	FELLING COST	BUNCHING HOURS	BUNCHING COST	YARDING HOURS	YARDING COST	LANDING HOURS	LANDING COST
1	174.56	7994.33	560.98	18159.91	613.64	27613.68	178.28	19853.21	178.28	15072.55
2	122.01	13521.84	894.33	58131.23	1052.15	52607.45	144.27	20918.67	144.27	15869.33

(TOTAL HOURS & COSTS)

UNIT#	AREA (ACRES)	(TONS) VOLUME	FELLING HOURS	FELLING COST	BUNCHING HOURS	BUNCHING COST	YARDING HOURS	YARDING COST	LANDING HOURS	LANDING COST
1	174.56	7994.33	645.48	20077.91	662.44	29405.68	205.68	23443.21	196.03	16008.80
2	122.01	13521.84	958.33	60371.23	1161.65	57534.95	169.57	23448.67	167.27	16674.33

PRESS RETURN KEY TO CONTINUE

Next the user is given the opportunity to select "parameter summaries" from the following menu. Each of these is illustrated with examples.

ENTER THE APPROPRIATE NUMBER FOR PARAMETER SUMMARY

0=NONE
1=HOURS
2=COST
3=$/TON
4=TONS/HR
5=$/HR

SUMMARY FOR HOURS (%)
(PRODUCTIVE HOURS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	560.98 (37.%)	613.64 (40.%)	178.28 (12.%)	178.28 (12.%)	1531.18 (100%)
2	%	894.33 (40.%)	1052.15 (47.%)	144.27 (6.%)	144.27 (6.%)	2235.02 (100%)
TOTAL	%	1455.31 (39.%)	1665.78 (44.%)	322.55 (9.%)	322.55 (9.%)	

NON-PRODUCTIVE(SETUP) HOURS

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	84.50 (47.%)	48.80 (27.%)	27.40 (15.%)	17.75 (10.%)	178.45 (100%)
2	%	64.00 (29.%)	109.50 (49.%)	25.30 (11.%)	23.00 (10.%)	221.80 (100%)
TOTAL	%	148.50 (37.%)	158.30 (40.%)	52.70 (13.%)	40.75 (10.%)	
COMBINED TOTAL	%	1603.81 (38.%)	1824.09 (44.%)	375.25 (9.%)	363.30 (9.%)	

SUMMARY FOR COSTS (%)

(PRODUCTIVE COSTS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	18159.92 (23.%)	27613.70 (34.%)	19853.21 (25.%)	15072.55 (19.%)	80699.39 (100%)
2	%	58131.22 (39.%)	52607.44 (36.%)	20918.67 (14.%)	15869.33 (11.%)	147526.7 (100%)
TOTAL	%	76291.14 (33.%)	80221.14 (35.%)	40771.88 (18.%)	30941.88 (14.%)	

(TOTAL PRODUCTIVE & SETUP COSTS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	20077.92 (23.%)	29405.70 (33.%)	23443.21 (26.%)	16008.80 (18.%)	88935.64 (100%)
2	%	60371.22 (38.%)	57534.94 (36.%)	23448.67 (15.%)	16674.33 (11.%)	158029.2 (100%)
TOTAL	%	80449.14 (33.%)	86940.64 (35.%)	46891.88 (19.%)	32682.00 (13.%)	

SUMMARY FOR $/TON (%)
(PRODUCTIVE COSTS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	2.27 (23.%)	3.45 (34.%)	2.48 (25.%)	1.89 (19.%)	10.09 (100%)
2	4.30 (39.%)	3.89 (36.%)	1.55 (14.%)	1.17 (11.%)	10.91 (100%)
WGHT. AVG.	3.55 (33.%)	3.73 (35.%)	1.89 (18.%)	1.44 (14.%)	

SUMMARY FOR TONS/HR (%)
(PRODUCTIVE HOURS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	14.25 (12.%)	13.03 (11.%)	44.84 (38.%)	44.84 (38.%)	116.96 (100%)
2	15.12 (7.%)	12.85 (6.%)	93.73 (44.%)	93.73 (44.%)	215.42 (100%)
WGHT. AVG.	14.78 (9.%)	12.92 (8.%)	66.71 (41.%)	66.71 (41.%)	

SUMMARY FOR $/HR (%)
(PRODUCTIVE HRS & COSTS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	32.37 (8.%)	45.00 (12.%)	111.36 (29.%)	84.54 (22.%)	390.24 (100%)
2	65.00 (11.%)	50.00 (9.%)	145.00 (25.%)	110.00 (19.%)	585.42 (100%)
WGHT. AVG.	52.42 (16.%)	48.16 (15.%)	126.40 (39.%)	95.93 (30.%)	

The model concludes with an "echo" of the input data that generated the various output options.

FELLING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	CYCLES	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	TREE WEIGHT (LBS)
ZONE(1)	3.57	7191.90	8.00	30.00	1.10	20.00	1140.53
ZONE(2)	3.57	2236.29	8.00	40.00	.80	35.00	3481.70

LOGGING UNIT (2)

	CYCLE TIME (MIN)	CYCLES	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	TREE WEIGHT (LBS)
ZONE(1)	3.57	15030.70	7.50	65.00	.50	35.00	1799.23

PRESS RETURN KEY TO CONTINUE

BUNCHING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	LOAD WEIGHT (LBS)
ZONE(1)	4.00	8.00	45.00	.50	35.00	9000.00
ZONE(2)	3.60	7.50	45.00	.80	40.00	8500.00

LOGGING UNIT (2)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	LOAD WEIGHT (LBS)
ZONE(1)	4.20	8.00	50.00	.75	45.00	10000.00

PRESS RETURN KEY TO CONTINUE

YARDING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	6.40	7.00	100.00	.90	150.00
ZONE(2)	5.30	7.50	125.00	1.00	110.00

LOGGING UNIT (2)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	3.20	7.50	145.00	1.10	100.00

PRESS RETURN KEY TO CONTINUE

LANDING INPUTS

LOGGING UNIT (1)

	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	80.00	.50	50.00
ZONE(2)	90.00	.75	55.00

LOGGING UNIT (2)

	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	110.00	1.00	35.00

CLEARWATER FOREST EXAMPLE

The Clearwater and Gallatin National Forests are quite different in the characteristics of their timber and terrain. In general, the Gallatin consists of steep terrain with much small timber (large quantities of lodgepole pine) while the Clearwater has more gentle terrain with much larger timber and a diversity of species. In fact, much of the terrain in the Jackknife Meadows area, shown in Figure 11, is amenable to tractor

skidding. However, this example will assume that the Helistat will yard tree-length timber from prebunched loads.

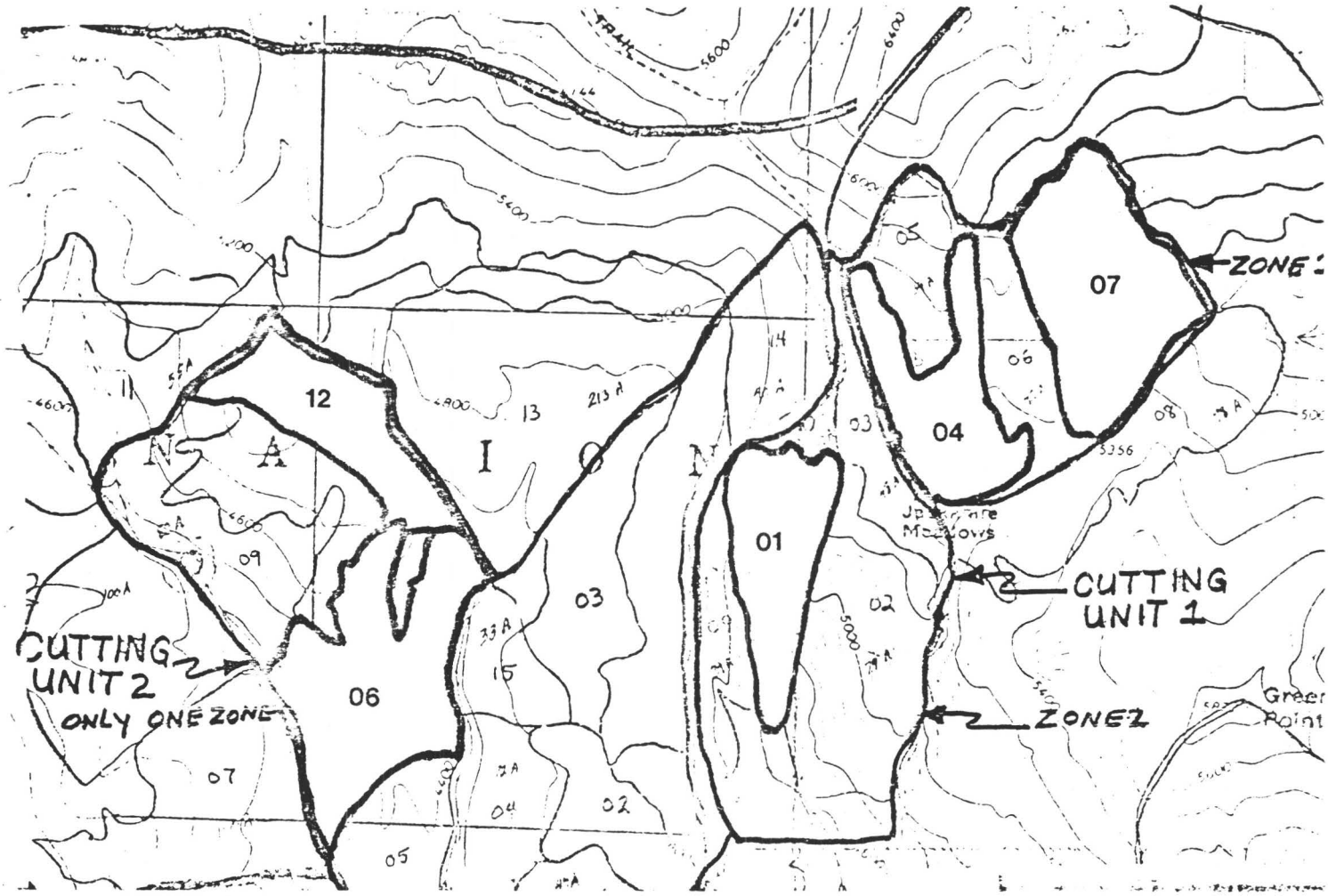


Figure 11. Clearwater Forest Example

This site consists of two cutting units. Cutting Unit 1 on the right is further partitioned into two zones. Within Zone 1 two stands will be harvested, 04 and 07. Notice that stands 05 and 06 will not be cut. Zone 2 in Cutting Unit 1 contains several stands, but only stand 01 will be harvested.

Cutting Unit 2 consists of a single zone from which stands 06 and 12 will be harvested.

Assumptions

Two crawler tractors will be used to prebunch trees into unit loads. The crew for each will consist of an operator and a helper. The tractors will cost $160,000 each and have an eight-year life. Each will operate about 1200 hours per year and require about one hour per day maintenance. Each tractor and crew will cost about $120 per hour to operate.

The system at the landing consists of two medium sized loaders with tracks, each of which costs about $289,000. The loaders will have a twenty percent salvage value after a life of eight years. Each loader will have a crew consisting of an operator and two helpers, and will cost about $171 per hour to operate.

Cost calculations for both the bunching and landing systems are documented in Appendix 4.

Timber Characteristic

The timber data for the Gallatin example was structured to represent a typical site. The Clearwater example, however, is based upon actual stand table data furnished by foresters. Timber inventories were entered into a database file as shown in Figure 12. Five stands of data are depicted with each stand beginning with #E4292 which defines the forest, ranger district, drainage, and sub-drainage. Each line of the file describes a particular set of timber in the stand. Appendix 1 describes how the data file was structured and what the numbers signify.

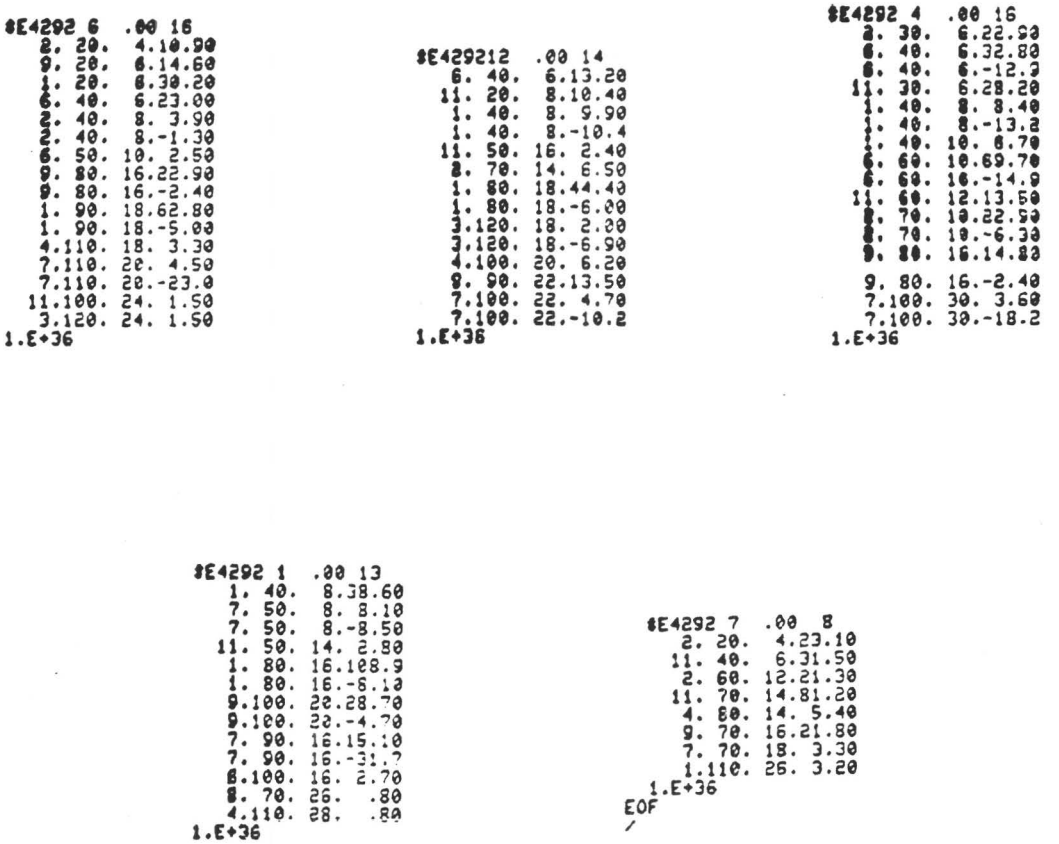


Figure 12. Timber Data File - Clearwater Example

Figure 13 indicates how the timber database information is stratified by the computer so that it conforms to the zone designations just described.

The Interactive Program

The interactive sequence of questions and responses will essentially be the same as in the Gallatin example. Therefore, less discussion of each step will be necessary in this example.

```
$E4292    UNIT 1    ZONE 1
      2.00      34.95       6.00      22.90
      6.00      44.95       6.00      32.80
      6.00      44.95       6.00     -12.90
     11.00      34.95       6.00      28.20
      1.00      44.95       8.00       8.40
      1.00      44.95       8.00     -13.20
      1.00      44.95      10.00       6.70
      6.00      64.95      10.00      69.70
      6.00      64.95      10.00     -14.90
     11.00      64.95      12.00      13.50
      2.00      74.95      10.00      22.90
      2.00      74.95      10.00      -6.30
      9.00      84.95      16.00      14.80
      9.00      84.95      16.00      -2.40
      7.00     104.95      30.00       3.60
      7.00     104.95      30.00     -18.20
      2.00      24.95       4.00      23.10
     11.00      44.95       6.00      31.50
      2.00      64.95      12.00      21.30
     11.00      74.95      14.00      81.20
      4.00      84.95      14.00       5.40
      9.00      74.95      16.00      21.80
      7.00      74.95      18.00       3.30
      1.00     114.95      26.00       3.20
THERE ARE24 ENTRIES IN THE DATA FILE FOR ZONE  1
$E4292    UNIT 1    ZONE 2
      1.00      44.95       8.00      38.60
      7.00      54.95       8.00       8.10
      7.00      54.95       8.00      -8.50
     11.00      54.95      14.00       2.80
      1.00      84.95      16.00     108.90
      1.00      84.95      16.00      -8.10
      9.00     104.95      20.00      28.70
      9.00     104.95      20.00      -4.70
      7.00      94.95      16.00      15.10
      7.00      94.95      16.00     -31.70
      6.00     104.95      16.00       2.70
      8.00      74.95      26.00        .80
      4.00     114.95      28.00        .80
THERE ARE13 ENTRIES IN THE DATA FILE FOR ZONE  2
```

```
$E4292    UNIT 2    ZONE 1
      2.00       3.00     124.95      18.00
      9.00       4.00     104.95      20.00
      1.00       9.00      94.95      22.00
      6.00       7.00     104.95      22.00
      2.00       7.00     104.95      22.00
      2.00      44.95       8.00      -1.30
      6.00      54.95      10.00       2.50
      9.00      84.95      16.00      22.90
      9.00      84.95      16.00      -2.40
      1.00      94.95      18.00      62.80
      1.00      94.95      18.00      -5.00
      4.00     114.95      18.00       3.30
      7.00     114.95      20.00       4.50
      7.00     114.95      20.00     -23.00
     11.00     104.95      24.00       1.50
      3.00     124.95      24.00       1.50
      6.00      44.95       6.00      15.20
     11.00      24.95       8.00      10.40
      1.00      44.95       8.00       9.90
      1.00      44.95       8.00     -10.40
     11.00      54.95      16.00       2.40
      2.00      74.95      14.00       6.50
      1.00      84.95      18.00      44.40
      1.00      84.95      18.00      -6.00
      3.00     124.95      18.00       2.00
      3.00     124.95      18.00      -6.90
      4.00     104.95      20.00       6.20
      9.00      94.95      22.00      15.50
      7.00     104.95      22.00       4.70
      7.00     104.95      22.00     -10.30
THERE ARE30 ENTRIES IN THE DATA FILE FOR ZONE  1
```

Figure 13. Timber Data Sorted by Cutting Unit and Zone

The model is initiated by asking the user to indicate the number of cutting units in the site (2 in this case). Next, the user is asked to indicate whether the area to be harvested is to be digitized from a map or if site parameters are to be input directly from the terminal. The map in Figure 11 was digitized as directed by the program. The following figures indicate sequentially how the map was digitized. Some steps have been omitted in the interest of brevity.

Digitizing Areas

First the map scale data is initialized.

```
TYPE 6 CHARACTER OUTPUT FILE NAME,THEN :125CLRH20:125
Type a title for the data file -> BUCKINGHAM LAKE
What is the X coordinate of lower left corner? 0.0
What is the Y coordinate of lower left corner? 0.0
What is the X coordinate of upper right corner? 13200.0
What is the Y coordinate of upper right corner? 13200.0
```

Next, the user is instructed to digitize the boundary of Cutting Unit 1, as shown in Figure 14.

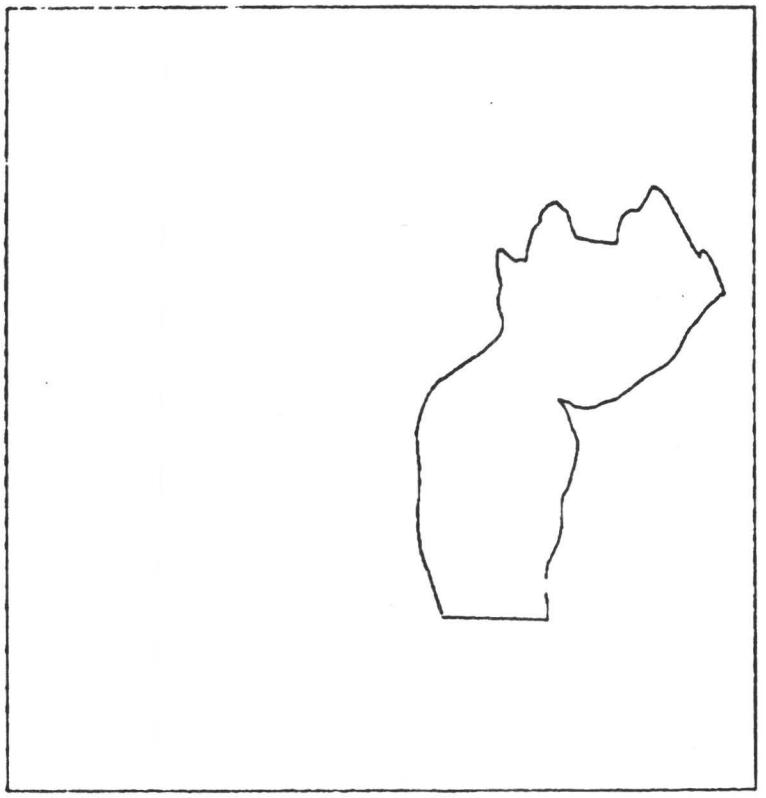


Figure 14 - Cutting Unit 1

Next, the user is given the opportunity to partition the cutting unit into zones to reflect different terrain and timber characteristics. Figure 15 shows Zones 1 and 2 after digitizing.

The user is then asked to input the stand number of the next stand to be digitized with Zone 1. In this case Stand 04 was input and digitized, followed by the same steps for Stand 07. Figure 16 indicates the results of these operations.

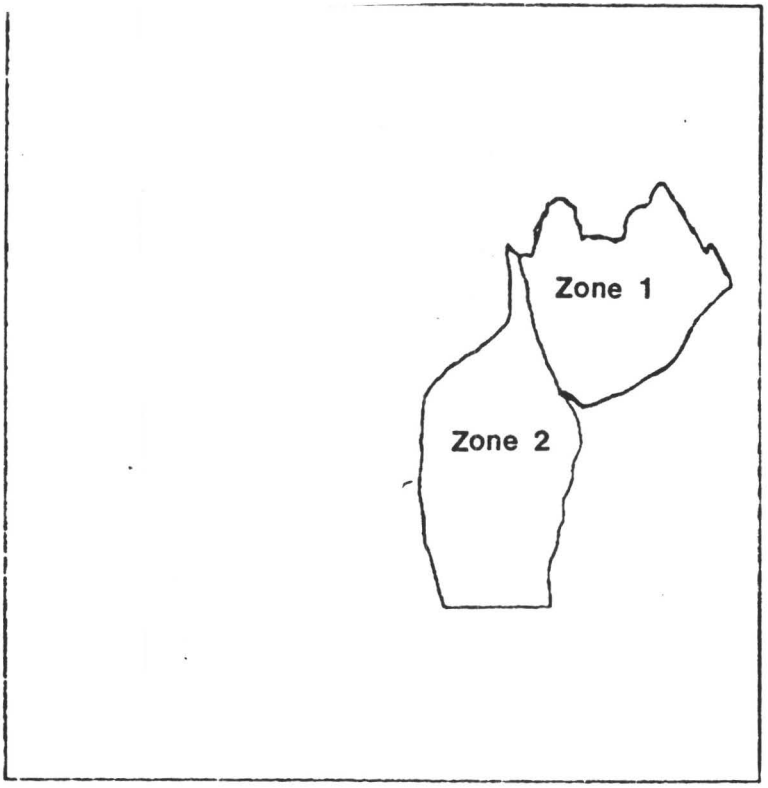


Figure 15. Zones 1 and 2

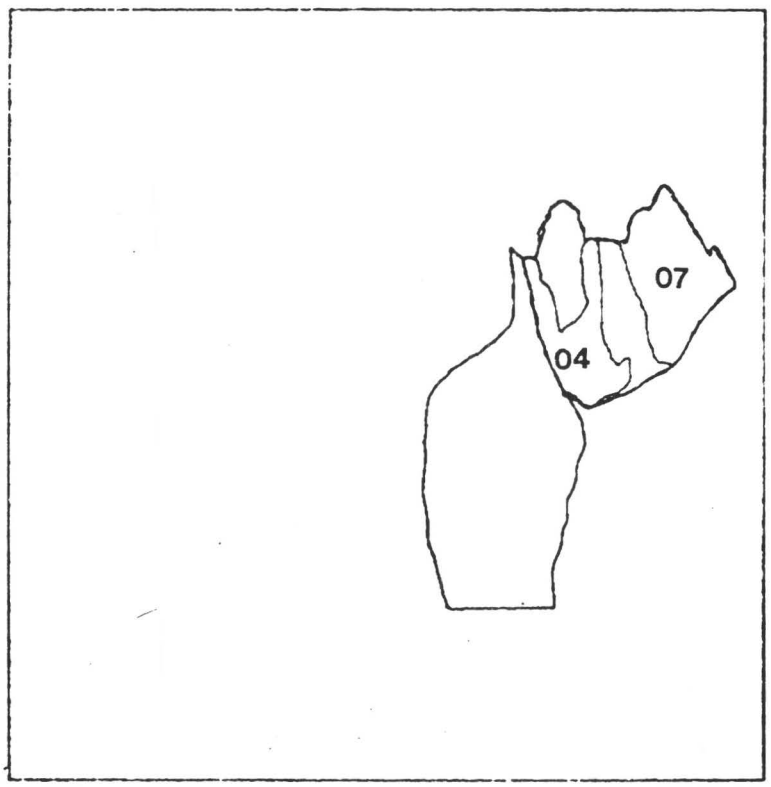


Figure 16. Stands 04 and 07

Since only two stands were included in Zone 1, the program asks similar questions regarding Zone 2 at this point. Figure 17 indicates the results of digitizing Stand 01 in Zone 2.

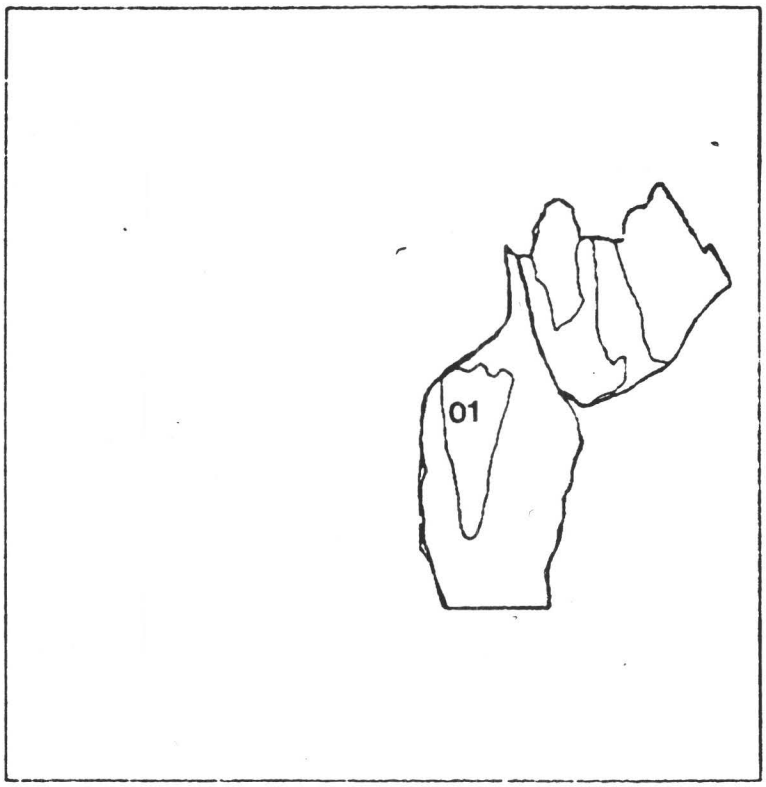


Figure 17. Stand 01 in Zone II

With all zones and stands within Cutting Unit 1 now digitized, the program repeats the same process for Cutting Unit 2 and the zones and stands within it. Figure 18 shows Cutting Unit 2 following digitization.

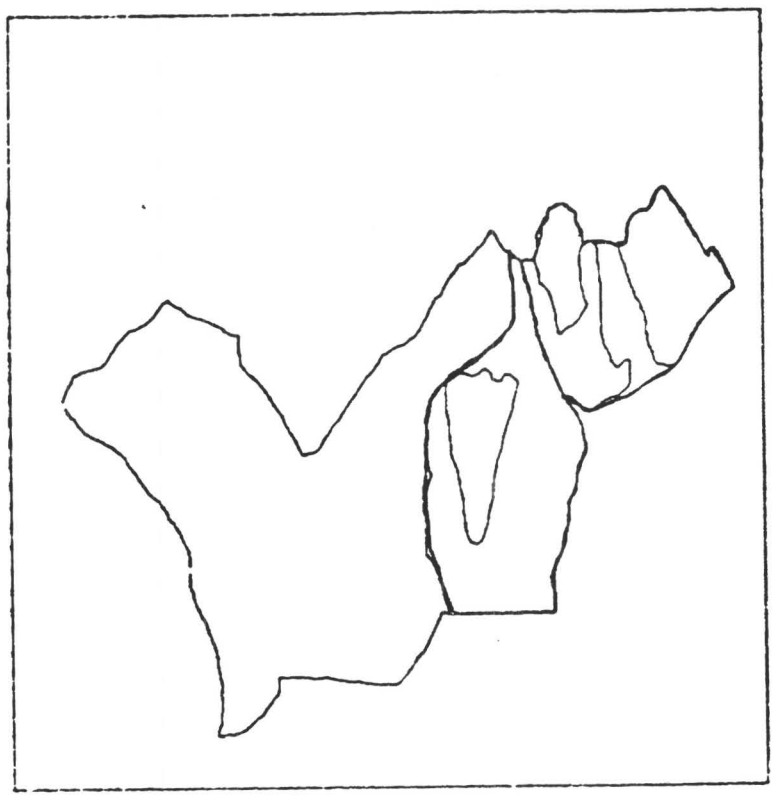


Figure 18. Cutitng Unit 2

Since only one zone is desired in this cutting unit, it was digitized next, followed by the stands within it as shown in Figure 19. When the last stand is digitized the program automatically calculates the areas of each digitized portion of land as shown. This information is stored and matched-up with the timber data in Figure 13 such that an accurate assessment can be made of the timber to be harvested and transported from the forest. Now the actual simulation can begin with the felling.

```
AREA OF CUTTING UNIT  1 -     445.0007
AREA OF ZONE          1 -     194.5056
AREA OF STAND         4 -      52.6564
AREA OF STAND         7 -      74.0045
AREA OF ZONE          2 -     250.2773
AREA OF STAND         1 -      51.6497


AREA OF CUTTING UNIT  2 -     677.7695
AREA OF ZONE          1 -     677.7695
AREA OF STAND         6 -      94.2184
AREA OF STAND        12 -      49.0592

PRESS THE RETURN KEY TO CONTINUE
```

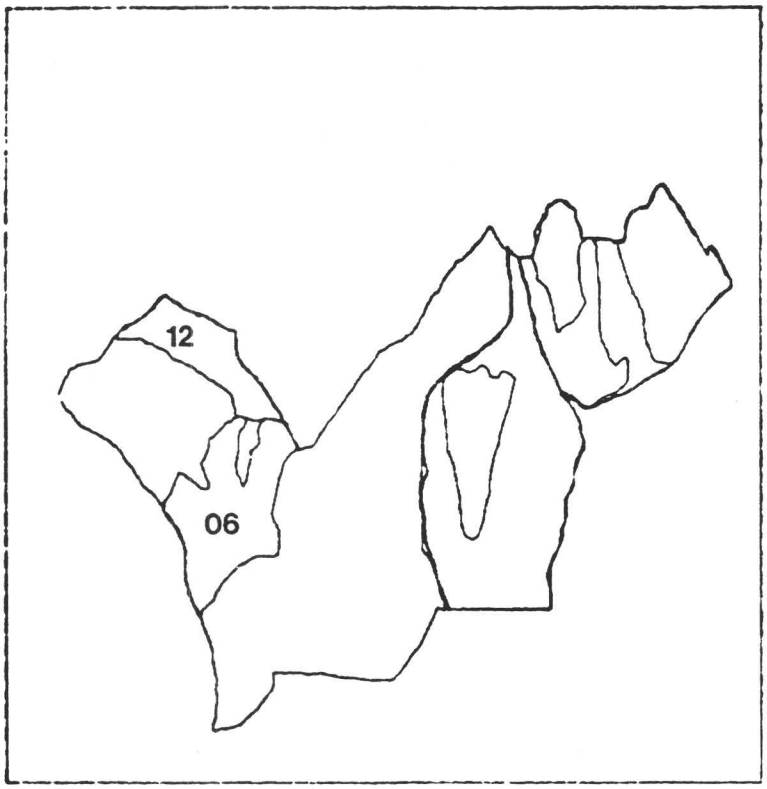


Figure 19. Stands 06 and 12

Felling

Figure 20 illustrates the interactive nature of the questions regarding the felling operations. Notice that after the user inputs the required data, the program retreives the timber data base information, calculates, and outputs the average tree size within each zone.

```
INPUT FELLING DATA

DO YOU WISH TO INPUT THE FELLING CYCLE TIMES OR USE THE DEFAULT VALUES FROM EMPIRICAL DATA ?
1 • INPUT MY VALUES        2 • USE DEFAULT VALUES
?
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 1•31.25
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 1 •.3
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 1 •0
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1 •8
SYSTEM FELLING COST/HR FOR ZONE 2 OF UNIT 1•31.25
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 2 OF UNIT 1 •.5
SYSTEM SET-UP COST ($/HR) FOR ZONE 2 OF UNIT 1 •0
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2 •7
SYSTEM FELLING COST/HR FOR ZONE 1 OF UNIT 2•31.25
THE NUMBER OF SYSTEM SET-UP HOURS PER DAY FOR ZONE 1 OF UNIT 2 •.4
SYSTEM SET-UP COST ($/HR) FOR ZONE 1 OF UNIT 2 •0
TOTAL WEIGHT(TONS) OF TREES IN ZONE 1 •       18727.0

AVERAGE TREE SIZE(LBS) IN ZONE 1 •       1677.8
PRESS RETURN KEY TO CONTINUE
TOTAL WEIGHT(TONS) OF TREES IN ZONE 2 •       15700.3

AVERAGE TREE SIZE(LBS) IN ZONE 2 •       3274.9
PRESS RETURN KEY TO CONTINUE
TOTAL WEIGHT(TONS) OF TREES IN ZONE 1 •       27366.0

AVERAGE TREE SIZE(LBS) IN ZONE 1 •       3043.2
PRESS RETURN KEY TO CONTINUE
```

Figure 20. Felling Input Data

Next optional output reports may be selected from the menu below.

```
ENTER NUMBER FOR OUTPUT DESIRED

0•NONE
1•PRODUCTIVITY AND COSTS BASED ON PER HOUR DATA
2•PRODUCTIVITY AND COSTS BASED ON PER DAY BASIS
```

Figure 21 shows the two reports available on felling activities.

FELLING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR
ZONE(1)	1328.26	18726.96	41508.27	2.22	14.10	31.25
ZONE(2)	570.50	15700.28	17828.15	1.14	27.52	31.25
TOTALS	1898.77	34427.25	59336.42	1.72	18.13	31.25

FELLING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR
ZONE(1)	1070.09	27365.98	33440.33	1.22	25.57	31.25
TOTALS	1070.09	27365.98	33440.33	1.22	25.57	31.25

PRESS RETURN KEY TO CONTINUE

FELLING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	7.70	.30	172.50	41508.27	.00	41508.27	18726.96
ZONE(2)	7.50	.50	76.07	17828.15	.00	17828.15	15700.28
TOTAL			248.57	59336.42	.00	59336.42	34427.25

FELLING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.60	.40	162.13	33440.33	.00	33440.33	27365.98
TOTAL			162.13	33440.33	.00	33440.33	27365.98

PRESS RETURN KEY TO CONTINUE

Figure 21. Output Reports on Felling

The number of trees felled, along with other data, is now passed on to the bunching section of the program where preparation is made to assemble unit loads for the Helistat.

Bunching

Two crawler tractors will be used to perform bunching operations. The cost input data is based upon data in Appendix 2. Figure 22 indicates the input data for the bunching system.

```
INPUT THE BUNCHING DATA


BUNCH CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 1•5
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1•8
SYSTEM BUNCHING COST/HR FOR ZONE 1 OF UNIT 1•241
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1•1
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 1•40
HELISTAT TARGET LOAD IN POUNDS•18000


BUNCH CYCLE TIME(MIN)FOR ZONE 2 OF UNIT 1•5.8
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1•8
SYSTEM BUNCHING COST/HR FOR ZONE 2 OF UNIT 1•245
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1•1.2
THE SYSTEM SETUP COST ($/HR) FOR ZONE 2 OF UNIT 1•40
HELISTAT TARGET LOAD IN POUNDS•17500


BUNCH CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 2•6.2
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2•8
SYSTEM BUNCHING COST/HR FOR ZONE 1 OF UNIT 2•2.441
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2•1.25
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 2•44
HELISTAT TARGET LOAD IN POUNDS•16200
```

Figure 22 - Bunching Input Data

Optional output reports concerning bunching are shown in Figure 23.

BUNCHING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	1860.31	18726.96	448335.7	23.94	10.07	241.00	2081.00
ZONE(2)	926.86	15700.28	227081.7	14.46	16.94	245.00	1795.00
TOTALS	2787.18	34427.25	675417.4	19.62	12.35	242.33	3876.00

BUNCHING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	1858.42	27365.98	447879.4	16.37	14.73	241.00	3379.00
TOTALS	1858.42	27365.98	447879.4	16.37	14.73	241.00	3379.00

PRESS RETURN KEY TO CONTINUE

BUNCHING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	7.00	1.00	265.76	448335.7	10640.00	458975.7	18726.96
ZONE(2)	6.80	1.20	136.30	227081.7	6576.00	233657.7	15700.28
TOTAL			402.06	675417.4	17216.00	692633.4	34427.25

BUNCHING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.75	1.25	275.32	447879.4	15180.00	463059.4	27365.98
TOTAL			275.32	447879.4	15180.00	463059.4	27365.98

PRESS RETURN KEY TO CONTINUE

Figure 23. Bunching Output Reports

Yarding

The Helistat yarding input data and output reports are shown in Figures 24 and 25 respectively.

```
INPUT YARDING DATA


YARDING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 1=6
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 1=8
SYSTEM YARDING COST/HR FOR ZONE 1 OF UNIT 1=800
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1=1.5
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 1=150


YARDING CYCLE TIME(MIN)FOR ZONE 2 OF UNIT 1=7.2
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 2 OF UNIT 1=8
SYSTEM YARDING COST/HR FOR ZONE 2 OF UNIT 1=800
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1=2.3
THE SYSTEM SETUP COST ($/HR) FOR ZONE 2 OF UNIT 1=150


YARDING CYCLE TIME(MIN)FOR ZONE 1 OF UNIT 2=8
THE NUMBER OF HOURS PER WORK DAY FOR ZONE 1 OF UNIT 2=8
SYSTEM YARDING COST/HR FOR ZONE 1 OF UNIT 2=800
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2=1.92
THE SYSTEM SETUP COST ($/HR) FOR ZONE 1 OF UNIT 2=150
```

Figure 24. Yarding Input Data

YARDING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	$ BUNCHES
ZONE(1)	208.10	18726.96	166480.0	8.89	89.99	800.00	2081.00
ZONE(2)	215.40	15700.28	172320.0	10.98	72.89	800.00	1795.00
TOTALS	423.50	34427.25	338800.0	9.84	81.29	800.00	3876.00

YARDING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	$ BUNCHES
ZONE(1)	450.53	27365.98	360426.7	13.17	60.74	800.00	3379.00
TOTALS	450.53	27365.98	360426.7	13.17	60.74	800.00	3379.00

PRESS RETURN KEY TO CONTINUE

YARDING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.50	1.50	32.02	166480.0	7425.00	173905.0	18726.96
ZONE(2)	5.70	2.30	37.79	172320.0	13110.00	185430.0	15700.28
TOTAL			69.80	338800.0	20535.00	359335.0	34427.25

YARDING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.08	1.92	74.10	360426.7	21600.00	382026.7	27365.98
TOTAL			74.10	360426.7	21600.00	382026.7	27365.98

PRESS RETURN KEY TO CONTINUE

Figure 25. Yarding Output Reports

Landing

The cost data for the landing system is included in Appendix 2. Figures 26 and 27 show the input data required and the output reports available, respectively.

```
INPUT THE LANDING DATA


SYSTEM LANDING COST/HR FOR ZONE 1 OF UNIT 1•344
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 1•2
THE SYSTEM SETUP COST PER HOUR FOR ZONE 1 OF UNIT 1•40


SYSTEM LANDING COST/HR FOR ZONE 2 OF UNIT 1•350
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 2 OF UNIT 1•1.8
THE SYSTEM SETUP COST PER HOUR FOR ZONE 2 OF UNIT 1•40


SYSTEM LANDING COST/HR FOR ZONE 1 OF UNIT 2•344
THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE 1 OF UNIT 2•2.1
THE SYSTEM SETUP COST PER HOUR FOR ZONE 1 OF UNIT 2•40
```

Figure 26. Landing Input Data

LANDING
(PRODUCTIVE HOURS)
LOGGING UNIT(1)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	208.10	18726.96	71586.41	3.82	89.99	344.00	2081.00
ZONE(2)	215.40	15700.28	75390.00	4.80	72.89	350.00	1795.00
TOTALS	423.50	34427.25	146976.4	4.27	81.29	347.05	3876.00

LANDING
(PRODUCTIVE HOURS)
LOGGING UNIT(2)

	HOURS	TONS	COST	$/TON	TONS/HR	$/HR	# BUNCHES
ZONE(1)	450.53	27365.98	154983.5	5.66	60.74	344.00	3379.00
TOTALS	450.53	27365.98	154983.5	5.66	60.74	344.00	3379.00

PRESS RETURN TO CONTINUE

LANDING
LOGGING UNIT(1)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.50	2.00	32.02	71586.41	2640.00	74226.41	18726.96
ZONE(2)	5.70	1.80	37.79	75390.00	2736.00	78126.00	15700.28
TOTAL			69.80	146976.4	5376.00	152352.4	34427.25

LANDING
LOGGING UNIT(2)

	TIME			COST (DOLLARS)			PRODUCTION (TONS)
	PRODUCTIVE (HR/DAY)	SETUP (HR/DAY)	TOTAL (DAYS)	PRODUCTIVE	SETUP	TOTAL	
ZONE(1)	6.08	2.10	74.10	154983.5	6300.00	161283.5	27365.98
TOTAL			74.10	154983.5	6300.00	161283.5	27365.98

PRESS RETURN TO CONTINUE

Figure 27. Landing Output Reports

ADDITIONAL REPORTS

At this point the simulation of the system is complete. The user is now offered several "overall" productivity and cost report options. For the Clearwater example, Figure 28 shows the Unit Activity Report and the Unit Summary Report.

UNIT ACTIVITY SUMMARY

(PRODUCTIVE HOURS & COSTS)

UNIT#	AREA (ACRES)	(TONS) VOLUME	FELLING HOURS	FELLING COST	BUNCHING HOURS	BUNCHING COST	YARDING HOURS	YARDING COST	LANDING HOURS	LANDING COST
1	178.21	34427.24	1898.77	59336.42	2787.18	675417.4	423.50	338500.0	423.50	146976.4
2	143.28	27365.98	1070.09	33440.32	1858.42	447879.2	450.53	360426.7	450.53	154983.5

(TOTAL HOURS & COSTS)

UNIT#	AREA (ACRES)	(TONS) VOLUME	FELLING HOURS	FELLING COST	BUNCHING HOURS	BUNCHING COST	YARDING HOURS	YARDING COST	LANDING HOURS	LANDING COST
1	178.21	34427.24	1989.17	59336.42	3217.58	692633.4	560.40	359335.0	557.90	152352.4
2	143.28	27365.98	1135.29	33440.32	2203.42	463059.2	594.53	382026.7	608.03	161283.5

PRESS RETURN KEY TO CONTINUE

UNIT SUMMARY REPORT

		HOURS	TONS	COST	$/TON
UNIT(1)	PRODUCTIVE	5532.95	34427.25	1220530.	35.45
	SETUP	792.10		43127.00	
	TOTAL	6325.05	34427.25	1263657.	36.71
UNIT(2)	PRODUCTIVE	3829.57	27365.98	996729.9	36.42
	SETUP	711.70		43080.00	
	TOTAL	4541.27	27365.98	1039810.	38.00
SITE TOTALS	PRODUCTIVE	9362.52	61793.23	2217260.	35.88
	SETUP	1503.80		86207.00	
	TOTAL	10866.32	61793.23	2303467.	37.28

Figure 28. "Overall" Site Reports Available

Next the user is given the opportunity to select "parameter summaries" from the menu shown below. Each of these is illustrated with examples.

ENTER THE APPROPRIATE NUMBER FOR PARAMETER SUMMARY

0=NONE
1=HOURS
2=COST
3=$/TON
4=TONS/HR
5=$/HR

SUMMARY FOR HOURS (%)
(PRODUCTIVE HOURS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	1898.77 (34.%)	2787.18 (50.%)	423.50 (8.%)	423.50 (8.%)	5532.95 (100%)
2	%	1070.09 (28.%)	1858.42 (49.%)	450.53 (12.%)	450.53 (12.%)	3829.57 (100%)
TOTAL	%	2968.86 (32.%)	4645.60 (50.%)	874.03 (9.%)	874.03 (9.%)	

NON-PRODUCTIVE(SETUP) HOURS

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	90.40 (11.%)	430.40 (54.%)	136.90 (17.%)	134.40 (17.%)	792.10 (100%)
2	%	65.20 (9.%)	345.00 (48.%)	144.00 (20.%)	157.50 (22.%)	711.70 (100%)
TOTAL	%	155.60 (10.%)	775.40 (52.%)	280.90 (19.%)	291.90 (19.%)	
COMBINED TOTAL	%	3124.46 (29.%)	5421.00 (50.%)	1154.93 (11.%)	1165.93 (11.%)	

SUMMARY FOR COSTS (%)

(PRODUCTIVE COSTS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	59336.42 (5.%)	675417.4 (55.%)	339800.0 (28.%)	146976.4 (12.%)	1220530. (100%)
2	%	33440.33 (3.%)	447879.4 (45.%)	360426.7 (36.%)	154983.5 (16.%)	996729.9 (100%)
TOTAL	%	92776.75 (4.%)	1123297. (51.%)	699226.7 (32.%)	301959.9 (14.%)	

(TOTAL PRODUCTIVE & SETUP COSTS)

UNIT	%	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1	%	59336.42 (5.%)	692633.4 (55.%)	359335.0 (28.%)	152352.4 (12.%)	1263657. (100%)
2	%	33440.33 (3.%)	463059.4 (45.%)	382026.7 (37.%)	161283.5 (16.%)	1039810. (100%)
TOTAL	%	92776.75 (4.%)	1155693. (50.%)	741361.7 (32.%)	32767.00 (1.%)	

SUMMARY FOR $/TON (%)
(PRODUCTIVE COSTS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1 $	1.72 (5.%)	19.62 (55.%)	9.84 (28.%)	4.27 (12.%)	35.45 (100%)
2 $	1.22 (3.%)	16.37 (45.%)	13.17 (36.%)	5.66 (16.%)	36.42 (100%)
WGHT. AVG. $	1.50 (4.%)	18.18 (51.%)	11.32 (32.%)	4.89 (14.%)	

PRESS RETURN KEY TO CONTINUE

SUMMARY FOR TONS/HR (%)
(PRODUCTIVE HOURS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1 $	18.13 (9.%)	12.35 (6.%)	81.29 (42.%)	81.29 (42.%)	193.07 (100%)
2 $	25.57 (16.%)	14.73 (9.%)	60.74 (38.%)	60.74 (38.%)	161.78 (100%)
WGHT. AVG. $	20.81 (12.%)	13.30 (8.%)	70.70 (40.%)	70.70 (40.%)	

PRESS RETURN KEY TO CONTINUE

SUMMARY FOR $/HR (%)
(PRODUCTIVE HRS & COSTS ONLY)

UNIT #	FELLING	BUNCHING	YARDING	LANDING	TOTAL
1 $	31.25 (2.%)	242.33 (15.%)	800.00 (50.%)	347.05 (22.%)	1613.70 (100%)
2 $	31.25 (2.%)	241.00 (15.%)	800.01 (51.%)	344.00 (22.%)	1578.04 (100%)
WGHT. AVG. $	31.25 (2.%)	241.80 (17.%)	800.00 (56.%)	345.48 (24.%)	

PRESS RETURN KEY TO CONTINUE

ECHO CHECK

The model concludes with an "echo" of the input data that generated the various output reports.

FELLING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	CYCLES	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	TREE WEIGHT (LBS)
ZONE(1)	3.57	22323.77	8.00	31.25	.30	.00	1677.76
ZONE(2)	3.57	9588.25	8.00	31.25	.50	.00	3274.90

LOGGING UNIT (2)

	CYCLE TIME (MIN)	CYCLES	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	TREE WEIGHT (LBS)
ZONE(1)	3.57	17984.71	7.00	31.25	.40	.00	3043.25

BUNCHING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	LOAD WEIGHT (LBS)
ZONE(1)	5.00	8.00	241.00	1.00	40.00	18000.00
ZONE(2)	5.80	8.00	245.00	1.20	40.00	17500.00

LOGGING UNIT (2)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR	LOAD WEIGHT (LBS)
ZONE(1)	6.20	8.00	241.00	1.25	44.00	16200.00

YARDING INPUTS

LOGGING UNIT (1)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	6.00	8.00	800.00	1.50	150.00
ZONE(2)	7.20	8.00	800.00	2.30	150.00

LOGGING UNIT (2)

	CYCLE TIME (MIN)	WORK HR/DAY	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	8.00	8.00	800.00	1.92	150.00

LANDING INPUTS

LOGGING UNIT (1)

	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	344.00	2.00	40.00
ZONE(2)	350.00	1.80	40.00

LOGGING UNIT (2)

	PROD COST/HR	STUP HR/DAY	STUP COST/HR
ZONE(1)	344.00	2.10	40.00

APPENDICES

APPENDIX 1

TIMBER DATA FILES

In order to use the Front-End Model for assessing the timber on each area to be harvested, it is necessary to build the timber data base files prior to using them in simulation. An interactive program has been written to build the files for the user as he responds to questions.

BSDAT

BSDAT is an acronym for "build stand data file". A listing follows a discussion of how it works.

Since each stand possesses different timber characteristics and has a different location, each must have a unique identity. This is established by responding to the following questions.

ENTER THE NUMBER OF DIGITS INDICATED

FOREST NUMBER	=	_____ _____
BLOCK OR RANGER DISTRICT NUMBER	=	_____
COMPARTMENT OR MAIN DRAINAGE NUMBER	=	_____ _____
SUBCOMPARTMENT OR SUBDRAINAGE NUMBER	=	_____
STAND NUMBER	=	_____ _____
NUMBER OF ACRES IN THE STAND	=	_____ _____ _____
NUMBER OF NON-ZERO CELLS IN STAND EXAM	=	_____

This information will become a header entry for each file and will be the first record of the file. Next the program is ready to receive the timber data for the file.

```
ENTER CODES FOR SPECIE, HEIGHT, DBH, AND TREES
PER ACRE SEPARATED BY COMMAS AS FOLLOWS:
     1,80,14,317
INPUT 1,1,1,1 AS THE LAST DATA ENTRY

DO YOU WISH TO SEE A LIST OF INPUT CODES?
     Y=YES N=NO
```

A "Y" response causes the following to be printed.

```
ACCEPTABLE CODES

SPECIE                HEIGHT              DBH              TREES/ACRE

1-GRAND FIR           20-20-29.9          2-1,2.9          XXX.XX IF TREES ARE LIVE
2-SUBALPINE FIR       30-30-39.9          4-3-4.9          -XXX.XX IF TREES ARE DEAD
3-LARCH               40-40-49.9          6-5-6.9
4-ENGLE SPRUCE        50-50-59.9          8-7-8.9
5-WT.BARK PINE        60-60-69.9          10-9-10.9
6-LODGEPOLE PINE      70-70-79.9          12-11-12.9
7-W. WHITE PINE       80-80-89.9          14-13-14.9
8-PONDEROSA PINE      90-90-99.9          16-15-16.9
9-DOUG FIR            100-100-109.9       18-17-18.9
10-W. HEMLOCK         110-110-119.9       20-19-20.9
11-MT. HEMLOCK        120-120-129.9       22-21-22.9
12-LIMBER PINE        130-130-139.9       24-23-24.9
13-OTHER              140-140-149.9       CONTINUES THRU 40-40-

OBTAIN HARD COPY NOW FOR LATER REFERENCE IF DESIRED--PRESS RETURN KEY TO CONTINUE
```

According to these codes 1,80,14,317 would indicate that this stand contains 317 live grand fir trees per acre which are between 80 and 89.9 feet high, with a DBH of 13-14.9 inches. If the trees were dead, 317 would have been preceded by a minus sign.

The user then proceeds to enter the data describing the stand timber. For example,

```
6,70,10,37
6,60,9,45
1,90,18,22
1,1,1,1
```

where the last entry indicates that the file is complete.

Next, an "echo check" of the input data is printed so that the user can verify the entries. If the number of entries actually entered and the number of non-zero cells are different, an error message is printed. Then the user is given this message:

```
PLEASE VERIFY AND CORRECT ANY ERRONEOUS
ENTRIES BY SELECTING FROM THE MENU

                *****MENU*****

     1 = DELETE A CELL
     2 = ADD A CELL
     3 = MODIFY A CELL
     4 = NO DATA CHANGES REQUIRED
     5 = MODIFY HEADER INFORMATION ONLY
```

Responding with the proper code will allow modification or continuation of the program. The user is then prompted with this question.

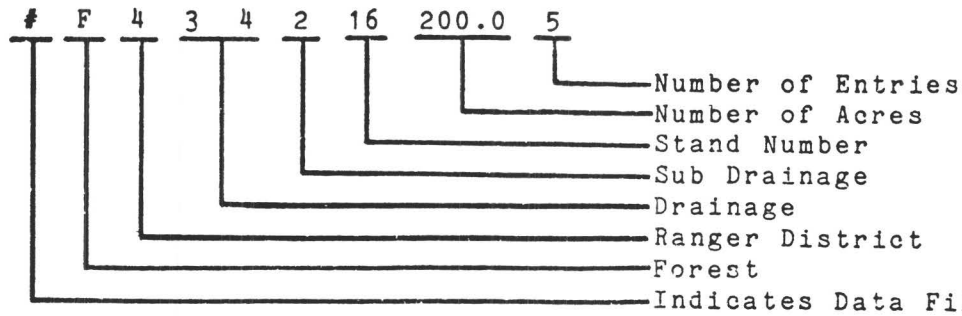
```
DO YOU DESIRE TO INPUT DATA FROM ANOTHER
STAND?
     Y=YES  N=NO
```

Responding with "Y" causes the program to start again, while a negative response terminates the program.

The following is an example of a file containing three stands of data. The 1.E+36 which terminates each set of stand data is a large number placed in the file by the program to indicate an end of file.

```
 $F434216200.0  5
    6. 79. 10.10.0
    6. 87. 12.12.0
    6. 83. 12.17.0
    6. 80. 13.21.0
    6. 90. 14.3.0
 1.E+36
 $F434213100.0  3
    6.100. 15.14.0
    1.110. 13.20.00
    6. 90.  9.17.0
 1.E+36
 $F434220l:0.0  4
    1. 70. 10. 50.
    4. 80.  9. 10.
    6. 92. 16. 30.
    6. 75. 14. 12.
 1.E+36
EOF
```

Notice that each stand contains a header file. The first header file is decoded as follows:



This program is "user friendly" and facilitates the structuring of timber data files for use with the Front-End Model.

The following is the computer code for BSDAT.

```
&BSDAT T=00004 IS ON CR00025 USING 00033 BLKS R=0088

0001  FTN4
0002        PROGRAM BSDAT
0003  C  THIS PROGRAM BUILDS DATA FILES FOR STANDS
0004  C  FIRST, IDENTIFY THE STAND EXPLICITLY
0005        COMMON ITEK(45)
0006        INTEGER FOR,RD,DR,SD
0007        DIMENSION ACRE(99),S(50),H(50),D(50),T(50),NAME(3),IFOR(20),
0008       1NUMB(10)
0009  C  IFOR(J) & NUMB(I) ARE ARRAYS FOR TABLE LOOKUPS USED IN HEADER INFO
0010        DATA IFOR/2HA ,2HB ,2HC ,2HD ,2HE ,2HF ,2HG ,2HH ,2HI ,
0011       12HJ ,2HK ,2HL ,2HM ,2HN ,2HO ,2HP ,2HQ ,2HR ,2HS ,2HT /
0012        DATA NUMB/2H0 ,2H1 ,2H2 ,2H3 ,2H4 ,2H5 ,2H6 ,2H7 ,
0013       12H8 ,2H9 /
0014        CALL INITT(45)
0015        CALL CHSIZ(3)
0016        L=0
0017        M=0
0018        MJ=99
0019     78 WRITE(9,50)
0020     50 FORMAT(1X,'********************PROGRAM BSDAT********************')
0021        WRITE(9,51)
0022     51 FORMAT(//,1X,'THIS PROGRAM BUILDS A STAND DATABASE')
0023     62 WRITE(9,1)
0024     1  FORMAT(1X,'ENTER THE NUMBER OF DIGITS INDICATED',//,
0025       11X,'FOREST NUMBER=__ ')
0026        READ(9,*)FOR
0027        WRITE(9,2)
0028     2  FORMAT(1X,'BLOCK OR RANGER DISTRICT NUMBER=_ ')
0029        READ(9,*)RD
0030        WRITE(9,3)
0031     3  FORMAT(1X,'COMPARTMENT OR MAIN DRAINAGE NUMBER=__ ')
0032        READ(9,*)DR
0033        WRITE(9,4)
0034     4  FORMAT(1X,'SUBCOMPARTMENT OR SUBDRAINAGE NUMBER=_ ')
0035        READ(9,*)SD
0036        WRITE(9,5)
0037     5  FORMAT(1X,'STAND NUMBER =__ ')
0038        READ(9,*)IST
0039        WRITE(9,6)
0040     6  FORMAT(1X,'NUMBER OF ACRES IN STAND=___.__ ')
0041        READ(9,*)ACRE(IST)
0042        WRITE(9,7)
0043     7  FORMAT(//,'THE NUMBER OF NON-ZERO CELLS IN STAND EXAM=__ ')
0044        READ(9,*)NC
0045        IF(L.EQ.5)GO TO 63
0046        CALL NEWPG
0047        WRITE(9,10)
0048     10 FORMAT(1X,'ENTER CODES FOR SPECIE,HEIGHT,DBH, AND TREES PER '
0049       2,'ACRE SEPARATED BY COMMAS AS FOLLOWS:',//,29X,
0050       3  '1,80,14,317',//,1X,'INPUT 1,1,1,1 AS THE LAST DATA ENTRY')
0051        WRITE(9,8)
0052     8  FORMAT(//,1X,'DO YOU WISH TO SEE A LIST OF THE INPUT CODES?
```

```
0053         4  Y=YES, N=NO')
0054          READ(9,48)II
0055    48    FORMAT(A1)
0056          IF(II.EQ.2HN )GO TO 49
0057          CALL NEWPG
0058          WRITE(9,11)
0059    11    FORMAT(1X,'ACCEPTABLE CODES',//)
0060          WRITE(9,12)
0061    12    FORMAT(1X,'SPECIE',15X,'HEIGHT',11X,'DBH',12X
0062         5,'TREES/ACRE',//)
0063          WRITE(9,13)
0064          WRITE(9,14)
0065          WRITE(9,15)
0066          WRITE(9,16)
0067          WRITE(9,17)
0068          WRITE(9,18)
0069          WRITE(9,19)
0070          WRITE(9,20)
0071          WRITE(9,21)
0072          WRITE(9,22)
0073          WRITE(9,23)
0074          WRITE(9,24)
0075          WRITE(9,25)
0076    13    FORMAT(1X,'1=GRAND FIR',9X,'20=20-29.9',7X,'2=1,2.9'
0077         6,9X,'XXX.XX IF TREES ARE LIVE')
0078    18    FORMAT(1X,'6=LODGEPOLE PINE',4X,'70=70-79.9',7X,'12=11-12.9')
0079    14    FORMAT(1X,'2=SUBALPINE FIR',5X,'30=30-39.9',7X,'4=3-4.9'
0080         7,8X,'-XXX.XX IF TREES ARE DEAD')
0081    19    FORMAT(1X,'7=W. WHITE PINE',5X,'80=80-89.9',7X,'14=13-14.9')
0082    15    FORMAT(1X,'3=LARCH',13X,'40=40-49.9',7X,'6=5-6.9')
0083    20    FORMAT(1X,'8=PONDEROSA PINE',4X,'90=90-99.9',7X,'16=15-16.9')
0084    16    FORMAT(1X,'4=ENGLE SPRUCE',6X,'50=50-59.9',7X,'8=7-8.9')
0085    21    FORMAT(1X,'9=DOUG FIR',10X,'100=100-109.9',4X,'18=17-18.9')
0086    17    FORMAT(1X,'5=WT.BARK PINE',6X,'60=60-69.9',7X,'10=9-10.9')
0087    22    FORMAT(1X,'10=W. HEMLOCK',7X,'110=110-119.9',4X,'20=19-20.9')
0088    23    FORMAT(1X,'11=MT. HEMLOCK',6X,'120=120-129.9',4X,'22=21-22.9')
0089    24    FORMAT(1X,'12=LIMBER PINE',6X,'130=130-139.9',4X,'24=23-24.9')
0090    25    FORMAT(1X,'13=OTHER',12X,'140=140-149.9',4X,'CONTINUES THRU 40='
0091         2'40+')
0092          WRITE(9,27)
0093    27    FORMAT(///,1X,'OBTAIN HARD COPY NOW FOR LATER REFERENCE IF '
0094         8,'DESIRED--PRESS RETURN KEY TO CONTINUE')
0095          CALL TINPT(ICHAR)
0096    49    CALL NEWPG
0097          WRITE(9,10)
0098    28    DO 30 K=1,9999
0099          WRITE(9,29)
0100    29    FORMAT(1X,'(SPECIE,HEIGHT,DBH,TREES/ACRE)=_,_,_,_ ')
0101          READ(9,*)S(K),H(K),D(K),T(K)
0102          NO=K-1
0103          G=S(K)+H(K)+D(K)+T(K)
0104          IF(G.EQ.4.)GO TO 40
0105    30    CONTINUE
0106    C  ECHO INPUT DATA FOR VERIFICATION
```

```
0107   40    CALL NEWPG
0108         S(K)=0
0109         H(K)=0
0110         T(K)=0
0111         D(K)=0
0112   61    WRITE(9,31)
0113   31    FORMAT(10X,'ECHO CHECK TO VERIFY INPUT DATA',//)
0114         WRITE(9,32)FOR
0115   32    FORMAT(10X,'FOREST NUMBER=',F2.0)
0116         WRITE(9,33)RD
0117         WRITE(9,34)DR
0118         WRITE(9,35)SD
0119         WRITE(9,36)IST
0120         WRITE(9,37)ACRE(IST)
0121         WRITE(9,38)NC
0122   33    FORMAT(10X,'RANGER DISTRICT NUMBER=',I2)
0123   34    FORMAT(10X,'COMPARTMENT(DRAINAGE)=',I2)
0124   35    FORMAT(10X,'SUBCOMPARTMENT(SUBDRAINAGE)=',I2)
0125   36    FORMAT(10X,'STAND NUMBER=',I2)
0126   37    FORMAT(10X,'ACRES IN THE STAND=',F5.2)
0127   38    FORMAT(10X,'NUMBER OF NON-ZERO CELLS OF DATA=',I2,//)
0128         WRITE(9,41)
0129   41    FORMAT(1X,'CELL NO',1X,'SPECIE',4X,'HEIGHT',4X,'DBH',7X,'TREES'
0130        1'/ACRE')
0131         DO 43 K=1,NO
0132         WRITE(9,42)K,S(K),H(K),D(K),T(K)
0133   42    FORMAT(3X,I3,3X,F4.0,7X,F4.0,4X,F4.0,11X,F5.2)
0134   43    CONTINUE
0135         IF(NC.NE.NO)GO TO 44
0136         GO TO 46
0137   44    WRITE(9,45)NC,NO
0138    45   FORMAT(1X,'WARNING--YOU INDICATED THE NUMBER OF NON-ZERO CELLS '
0139        1 'TO BE =',I3,/,1X,'BUT YOU ACTUALLY ENTERED',I3)
0140   46    WRITE(9,47)
0141   47    FORMAT(1X,'PLEASE VERIFY AND CORRECT ANY ERRONEOUS ENTRIES',
0142        1' BY SELECTING FROM THE MENU',///,22X,
0143        2'******MENU******',/,21X,
0144        3'1=DELETE A CELL',/,21X,
0145        4'2=ADD A CELL',/,21X,
0146        5'3=MODIFY A CELL',/,21X,
0147        6'4=NO DATA CHANGES REQUIRED',/,21X,
0148        7'5=MODIFY HEADER INFORMATION ONLY')
0149         READ(9,*)L
0150         IF(L.EQ.4)GO TO 55
0151         IF(L.EQ.5)GO TO 62
0152   52    WRITE(9,56)
0153   56    FORMAT(1X,'ENTER CELL NUMBER TO BE ALTERED_ ')
0154         READ(9,*)M
0155         GO TO(53,54,54)L
0156   53    S(M)=0.
0157         D(M)=0.
0158         T(M)=0.
0159         H(M)=0.
0160         GO TO 63
```

```
0161    54    WRITE(9,58)M
0162    58    FORMAT(1X,'ENTER SPECIE,HEIGHT,DBH,TREES/ACRE FOR',
0163         1 ' CELL NUMBER ',I3)
0164          READ(9,*)S(M),H(M),D(M),T(M)
0165          IF(L.EQ.2)NO=NO+1
0166    63    WRITE(9,59)
0167    59    FORMAT(//,1X,'HAVE ALL DATA CORRECTIONS BEEN MADE?  Y=YES N=NO')
0168          READ(9,60)NN
0169    60    FORMAT(A1)
0170          IF(NN.EQ.2HN )GO TO 46
0171          CALL NEWPG
0172          GO TO 61
0173    55    CONTINUE
0174    C  WRITE OUT DATA TO DISC FILE FOR LATER RETRIEVAL.
0175    C  HEADER INFORMATION**:FILE NAME WHICH IDENTIFIES FOREST,RANGER DIST,
0176    C  COMPARTMENT,AND SUB-COMPARTMENT(EX:#K6061=GALLATIN,RD=6,COMPART=06,
0177    C  SUBCOMPARTMENT=1).  THIS IS FOLLOWED BY STAND NO.,ACRES,AND THE
0178    C  NUMBER OF ENTRIES IN THE FILE.THEN SPECIE,HEIGHT,DBH, AND TREES
0179    C  PER ACRE FOLLOW.THE LAST ENTRY IS EOF MARKER 1.E36.
0180          ITEMP=IFOR(FOR)
0181          ITEMP=ITEMP/256.
0182          IPD=2H# /256
0183          IPD=IPD*256
0184          NAME(1)=IPD+ITEMP
0185          ITEMP=NUMB(RD+1)
0186          ITEMP=ITEMP/256
0187          ITEMP=ITEMP*256
0188          IDRT=(DR/10)+1
0189          ITMP=NUMB(IDRT)
0190          ITMP=ITMP/256
0191          NAME(2)=ITEMP+ITMP
0192          IX=(DR-((IDRT-1)*10))+1
0193          ITEMP=NUMB(IX)
0194          ITEMP=ITEMP/256
0195          ITEMP=ITEMP*256
0196          ITMP=NUMB(SD+1)
0197          ITMP=ITMP/256
0198          NAME(3)=ITEMP+ITMP
0199          IF(MJ.EQ.2HY )GO TO 79
0200          CALL SPOLU(LU,NAME,1,1,25)
0201     79   WRITE(9,75)(NAME(I),I=1,3),LU,NO
0202    75    FORMAT(1X,3A2,4X,I4,4X,I4)
0203          WRITE(LU,65)(NAME(I),I=1,3),IST,ACRE(IST),NO
0204    65    FORMAT(3A2,I2,F5.2,I3)
0205          DO 71 I=1,NO
0206          WRITE(LU,66)S(I),H(I),D(I),T(I)
0207    66    FORMAT(1X,3F4.0,F5.2)
0208    71    CONTINUE
0209          WRITE(LU,70)1.E36
0210    70    FORMAT(E6.1)
0211          WRITE(9,76)
0212    76    FORMAT(1X,'DO YOU DESIRE TO INPUT DATA FROM ANOTHER STAND?',
0213         1 '  Y=YES  N=NO')
0214          READ(9,77)MJ
```

```
0215   77   FORMAT(A1)
0216        IF(MJ.EQ.2HY )GO TO 78
0217        CALL SPOLU(LU,NAME,1,2,25)
0218        WRITE(9,80)
0219   80   FORMAT(1X,'ECHO CHECK--HOW MANY STANDS OF DATA ARE IN THIS FILE?'
0220        CALL SPOLU(LU,NAME,1,1,25)
0221        READ(9,*)NU
0222        DO 81 K=1,NU
0223        READ(LU,65)(NAME(I),I=1,3),IST,ACRE(IST),NO
0224        WRITE(9,65)(NAME(I),I=1,3),IST,ACRE(IST),NO
0225        DO 67 I=1,NO
0226        READ(LU,66)S(I),H(I),D(I),T(I)
0227        WRITE(9,66)S(I),H(I),D(I),T(I)
0228   67   CONTINUE
0229        READ(LU,70)XI
0230        WRITE(9,70)XI
0231   81   CONTINUE
0232        CALL SPOLU(LU,NAME,1,2,25)
0233        END
```

APPENDIX 2

CLEARWATER EQUIPMENT AND LABOR ESTIMATES

BUNCHING SYSTEM

Assumptions

- 2 crawler tractors for bunching ($160,000 each)
- interest rate = 12%
- crawlers operate about 1200 hours per year
- normal day is 7 hours productive operation
- maintenance requires 1 hour per day
- each crawler requires an operator and a helper
- 30,000 bd. ft. per day per tractor

Calculations per Machine

- Variable Costs:
  - Lubricants/year = (1200 hrs/yr)($.50/hr) = $ 600
  - Fuel/year = (5gal/hr)($1.20/gal)(1200 hrs/yr) = $7200
  - Operator/year = ($20/hr)(1200 hrs/yr) =$24,000
  - Helper/year = $16/hr)(1200 hrs/yr) =$19,200

    Total variable cost/year =$51,000
- Fixed Costs per Year:

  ($160,000-$32,000)(A/P, 12%,8)+$32,000(.12) =$29,607
- Overhead: ($8.81/1000 bd.ft.)

  (30,000 bd.ft./day) = $264/day

  ($264/day)(1200 hrs/yr/7 hrs/day) =$45,257

Total annual cost per crawler

= fixed cost + variable cost + overhead

= $29,607 + $51,000 + $45,257 =$125,864

. Profit and risk:

(.15)($125,864) =$18,880

. Grand Total Cost/Crawler/Year:

= $125,864 + $18,880 =$144,744

= ($144,744/1200 hrs/yr) =$120.62

LANDING SYSTEM

Assumptions

. 2 medium loaders with tracks required ($289,000 each)

. interest rate = 12%

. operational hours = 1200 per year

. one operator and 2 helps per machine

Calculations per Machine

. Variable Costs:

. Lubricants/year = (1200 hrs)($.30/hr) =$ 360

.Fuel/year= (5 gal/hr)($1.20/gal)(1200 hrs/yr)=$7,200

. Operator = ($20/hr)(1200 hrs/yr) =$24,000

. 2 helpers = 2($16/hr)(1200 hrs/yr) =$38,300

Total Variable Cost/Year =$69,960

. Fixed Cost per year:

($289,000-$57,800)(A/P,12%,8)+ $57,800(.12) =$57,477

. Overhead: $300/day

= ($300/day)(1200 hrs/yr/7hrs/day) =$51,428

Total Annual Cost Per Machine

= fixed cost + variable cost + overhead

= $57,471 + $69,960 + $51,428 =$178,865

. Profit and Risk:

(.15)($178,865) =$26,929

Grand Total Cost/Loader/Year

= $178,865 + $26,829 =$205,694

= ($205,694/1200hrs/yr) =$171.41

APPENDIX 3

DOCUMENTATION

The purpose of this appendix is to document the logic of the programs used to simulate the helistat logging system.

INTERATION GRAPHS

Iteration graphs have been constructed for each of the main subroutines or programs. The name of each program is given in capital letters and underlined. A brief description of the program follows the name. Next, each program called by another program is indicated by showing the underlined program name in a block that is indented. Branching is indicted by a triangle containing a statement within it. If the statement is true, the "THEN" side of the graph is followed. Otherwise, the "ELSE" side is excecuted.

The main program in this model is called HEST4. The simulation begins and ends within this program. HEST4 first calls program SORT which then determines whether the user desires to digitize areas and use actual stand data or prefers to enter estimates or averages directly from the terminal.

When coupled with the code itself and the variables identified via comment statements, the iteration graph provides the user with documentation of what each program does and how logic and control are transferred among them.

HEST4 - Main program - Controls other programs - Outputs reports.

SORT - Retrieves stand data from disc and sorts it into cutting units and zones.

FELL - Calculates cost and productivity data regarding felling activity.

BNCH - Calculates cost and productivity data regarding bunching activity.

YARD - Calculates cost and productivity data regarding yarding activity.

LAND - Calculates cost and productivity data regarding landing activity.

Optional output "Summary Report".
Optional output "Unit Activity Report".
Optional output "Parameter Summary".
  Based on: Hours, cost, cost per ton, tons per hour or cost per hour.
Echo check: Input values are output back to the user.
End.

SORT - Retrieves stand data from disc and sorts it into cutting units and zones.

If digitizing used for areas

Then

DIGBT - Digitizing program builds data file from digitizing tablet.

Else

Input from terminal:
Cutting units and areas.
Zones and areas.
Stands and areas.

Identify: Forest, Ranger District, . . Stand.

If digitizing and Stand table data used

Then

Calculate:
Number of trees per stand.
Accumulate per zone.
Convert to volume and weight per stand and zone.
Average tree weight.

Else

Return.

End.

DIGBT - Digitizing program - Builds data files from the digitizing tablet.

QUEST - Provides interactive questions while digitizing.

A2DBT - Calculates areas from data file.

End.

A2DBT - Calculates the areas of Units, Zones, and Stands.

Do for each Unit, Zone, and Stand.

Calculate the area using the formula:

AREA = (X1*Y2+X2*Y3+...Xn-1*Yn+Xn*Y1-Y1*X2-Y2*X3-...Yn-1*Xn-Yn*X1)/2

End do.

End.

FELL - Calculates cost and productivity data regarding felling activity.

If using empirical data

Then

Repeat.

Input:

- No. of Hours per work day,
- Felling system cost,
- Set up hours per day,
- Set up cost per day.

Until each zone is input.

Else

Repeat.

Input:

- Felling cycle time,
- No. of Cycles (trees) / zone,
- No. of Hours per work day,
- Set up cost per hour,
- Average tree size (lbs).

Until each zone is input.

Optional output reports on the following:

- Productivity and costs on a per hour basis.
- Productivity and costs on a per day basis.

End.

BNCH - Calculates cost and productivity data regarding bunching activity.

Repeat

Input:

Cycle time.

Number of hours per work day.

Bunching system cost per hour.

Set up hours per day.

Set up cost per hour.

HELISTAT TARGET LOAD (lbs).

Until each zone is input.

Optional output reports on bunching:

Productivity and costs on a per hour basis.

Productivity and costs on a per day basis.

End.

YARD - Calculates cost and productivity data regarding yarding activity.

Repeat.

Input:

Yarding cycle time.

Number of hours per work day.

Yarding system cost per hour.

Set up hours per day.

Set up cost per hour.

Until each zone is input.

Optional output reports on yarding:

Productivity and costs on a per hour basis.

Productivity and costs on a per day basis.

End.

LAND - Calculate cost and productivity data regarding landing activity.

Repeat.

Input:

System loading cost per hour.

Set up hours per day.

Set up cost per hour.

Until each zone is input.

End.

APPENDIX 4

This appendix contains all of the computer code required for running the Helistat Load Assembly Simulation Program.

```
&HEST4 T=00004 TS ON CR00025 USING 00092 BLKS R=0112

0001  FTN4
0002        PROGRAM HEST4
0003  C*************************************************************************
0004  C*********************************HELISTAT MODEL 1************************
0005  C*************************************************************************
0006  C
0007  CTHIS PROGRAM CALCULATES : TOTAL COST, TOTAL PRODUCTION, AND TOTAL TIME
0008  CTO HARVEST A SITE.
0009  C
0010        INTEGER FELL1(3),BNCH1(3),LAND1(3),YARD1(3),ZUNIT
0011        COMMON ITFK(45)
0012        COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
0013       1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
0014       2PHPDY(5,5),FELLTM(5,5),PR(5,5),DA
0015        INTEGER UNITS,DA,II(50),NZ(5,5)
0016        REAL AU(5),AZ(5,5),AS(5,5,20),AT(5)
0017        INTEGER COMMON(1),FELL(3),BNCH(3),YARD(3),LAND(3)
0018        EQUIVALENCE (COMMON(1),ITFK(1))
0019        DIMENSION T(10),P(10),C(10),CPTU(10),TPHU(10),CPHU(10)
0020       1,TTMU(10),PPRU(10),CCTU(10),CC(5,4),PP(5,4),TT(5,4),R(5,4),
0021       2DPT(5,4),TOT(5),FELLCT(5,5),BUNTM(5,5),BUNCT(5,5),
0022       3YARDCT(5,5),LANDCT(5,5),SUMT(4),PEC(4),WAVG(4),CSUM(4),
0023       4HSUM(4),PSUM(4),PHPD(5,5),SHPD(5,5),SCPH(5,5)
0024        DIMENSION HPWDF(5,5),PHPDF(5,5),SHPDF(5,5),SCPHF(5,5)
0025       1,HPWDB(5,5),PHPDB(5,5),SHPDB(5,5),SCPHB(5,5),HPWDY(5,5)
0026       2,SHPDY(5,5),SCPHY(5,5),SHPDL(5,5),SCPHL(5,5)
0027       3,HPWDL(5,5),PHPDL(5,5)
0028       4,TSHF(5),TSHB(5),TSHY(5),TSHL(5),TSCF(5),TSCB(5)
0029       5,TSCY(5),TSCL(5),TSHS(5),TSCS(5),PASH(5),PASC(5),CPTUU(5)
0030       6,PASHF(5),PASCF(5),PASHB(5),PASCB(5),PASHY(5),PASCY(5)
0031        DIMENSION PASHL(5),PASCL(5),Q(5),RR(5),V(5),E(5)
0032       1,PF(5),PB(5),PY(5),PL(5),DAYF(5,5),DAYB(5,5),DAYY(5,5),DAYL(5,5)
0033       2,TMDF(5,5),TMDB(5,5),TMDY(5,5),TMDL(5,5)
0034        DATA FELL/ 2HFE, 2HLL, 2H  /,
0035       &      BNCH/ 2HBN, 2HCH, 2H  /,
0036       &      YARD/ 2HYA, 2HRD, 2H  /,
0037       &      LAND/ 2HLA, 2HND, 2H  /
0038        DATA FELL1/ 2HFE,2HLL,2H# /,
0039       &BNCH1/ 2HBN,2HCH,2H# /,
0040       &YARD1/ 2HYA,2HRD,2H# /,
0041       &LAND1/ 2HLA,2HND,2H# /
0042  C
0043  CUSER INPUTS FOR SITE DATA: NO. OF UNITS, NO. OF ZONES(J) FOR UNIT(I),
0044  C UNITS= THE NUMBER OF UNITS PER SITE, ZUNIT(I)= THE NUMBER OF
0045  C ZONES PER UNIT.
0046  C
0047        CALL INITT(45)
0048        CALL CHSIZ(3)
0049        WRITE(9,710)
0050   710  FORMAT(1X,2("*****************************************",
0051       &"*****************************************",
0052       &"***************************",/,1X,))
```

```
0053          WRITE(9,711)
0054     711  FORMAT(1X,'*************************************    ',
0055         &'HELISTAT SYSTEMS SIMULATION MODEL    ****************',
0056         &'**********************',/)
0057          WRITE(9,710)
0058          WRITE(9,712)
0059     712  FORMAT(///,54X,'DEVELOPED BY',///)
0060          WRITE(9,714)
0061     713  FORMAT(51X,'USDA FOREST SERVICE',//,
0062         &37X,'INTERMOUNTAIN FOREST & RANGE EXPERIMENT STATION',//,
0063         &52X,'MISSOULA, MONTANA',///,58X,'1983',///)
0064          WRITE(9,713)
0065     714  FORMAT(49X,'COLLEGE OF ENGINEERING',//,
0066         &48X,'MONTANA STATE UNIVERSITY',//,
0067         &52X,'BOZEMAN,MONTANA',///,
0068         &50X,'IN COOPERATION WITH',//)
0069          WRITE(9,710)
0070          WRITE(9,710)
0071          WRITE(9,715)
0072     715  FORMAT(1X,////////,'PRESS THE RETURN KEY TO CONTINUE')
0073          CALL TINPT(ICHAR)
0074          CALL NEWPG
0075          WRITE(9,10)
0076       10 FORMAT(1X,'NUMBER OF UNITS (I) IN THE SITE ---> _')
0077          READ(9,*) UNITS
0078          NUNITS=UNITS
0079     C
0080      704  CALL SORT(FELLNC,PR,FELLTM,DA,AU,AZ,AS,II,NZ)
0081     C
0082     C FELLING : ACTIVITY 1
0083     C
0084       703 CALL SWAP (FELL)
0085     C
0086     C BUNCHING : ACTIVITY 2
0087     C
0088          CALL SWAP (BNCH)
0089     C
0090     C YARDING : ACTIVITY 3
0091     C
0092          CALL SWAP (YARD)
0093     C
0094     C UNLOADING : ACTIVITY 4
0095     C
0096          CALL SWAP (LAND)
0097     C
0098     C
0099          CALL SPOLU (LU(1),FELL1,1,1,25)
0100          CALL SPOLU (LU(2),BNCH1,1,1,25)
0101          CALL SPOLU (LU(3),YARD1,1,1,25)
0102          CALL SPOLU (LU(4),LAND1,1,1,25)
0103     C
0104     C  READ VARIABLES FROM LU'S
0105     C
0106          NUNITS=UNITS
```

```
0107         DO 637 I=1,NUNITS
0108         T(I)=0.
0109         C(I)=0.
0110         P(I)=0.
0111    637  CONTINUE
0112    C    CC(I,K),PP(I,K),TT(I,K)=COST,TONS,& TIME FOR UNIT I,ACTIVITY K
0113         DO 300 K=1,4
0114         DO 301 I=1,NUNITS
0115         READ(LU(K),71)TTMU(I),PPRU(I),CCTU(I)
0116         TT(I,K)=TTMU(I)
0117         PP(I,K)=PPRU(I)
0118         CC(I,K)=CCTU(I)
0119         T(I)=T(I)+TTMU(I)
0120         C(I)=C(I)+CCTU(I)
0121         IF(K.EQ.3)P(I)=PPRU(I)
0122    301  CONTINUE
0123    300  CONTINUE
0124         DO 400 I=1,NUNITS
0125         N=ZUNIT(I)
0126         DO 401 J=1,N
0127         READ(LU(1),140)FELLTM(I,J),FELLNC(I,J),FELLCT(I,J),FELLPR(I,J),
0128        1HPWDF(I,J),PHPDF(I,J),SHPDF(I,J),SCPHF(I,J),TMDF(I,J)
0129         READ(LU(2),141)BUNTM(I,J),BUNCT(I,J),TLOAD(I,J),HPWDB(I,J),
0130        1PHPDB(I,J),SHPDB(I,J),SCPHB(I,J),TMDB(I,J)
0131         READ(LU(3),142)YARDTM(I,J),YARDCT(I,J),HPWDY(I,J),PHPDY(I,J),
0132        1SHPDY(I,J),SCPHY(I,J),TMDY(I,J)
0133         READ(LU(4),143)LANDCT(I,J),SHPDL(I,J),SCPHL(I,J),HPWDL(I,J),
0134        1PHPDL(I,J),TMDL(I,J)
0135    140  FORMAT(9(F8.2,1X))
0136    141  FORMAT(8(F8.2,1X))
0137    142  FORMAT(7(F8.2,1X))
0138    143  FORMAT(6(F8.2,1X))
0139    401  CONTINUE
0140    400  CONTINUE
0141         CALL NEWPG
0142         TCOST=0.
0143         TPRD=0.
0144         TTIM=0.
0145         CTHR=0.
0146         TNHR=0.
0147         CTTN=0.
0148         DO 83 I=1,10
0149         TSCL(I)=0.
0150         TSCY(I)=0.
0151         TSCB(I)=0.
0152         TSHF(I)=0.
0153         TSHB(I)=0.
0154         TSHY(I)=0.
0155         TSHL(I)=0.
0156         TSCF(I)=0.
0157         TSHS(I)=0.
0158         TSCS(I)=0.
0159         PASH(I)=0.
0160         PASC(I)=0.
```

```
0161          CPTUU(I)=0.
0162    83    CONTINUE
0163    C
0164    C  SUMMARY REPORT FOR ANY UNIT DESIRED?
0165       43 WRITE(9,11)
0166       11  FORMAT(1X,"ENTER 77 IF A UNIT SUMMARY IS DESIRED; 99 FOR NONE",
0167          &" ---> _")
0168          READ(9,*)KK
0169          CALL NEWPG
0170          IF(KK-77)43,42,42
0171    C
0172    71    FORMAT(1X,10(F10.2,1X))
0173    C  T(I)=TIME FOR ALL ACTIVITIES IN UNIT I
0174    C  CALCULATE TOTAL SETUP HOURS AND COST BY UNIT
0175    C  TSHF(I),TSHB(I),TSHY(I),TSHL(I)=TOTAL SETUP HOURS,FELL,BNCH,ETC.
0176    C  TSCF(I),TSCB(I),TSCY(I),TSCL(I)=TOTAL SETUP $,FELL,BNCH,ETC.
0177    C  SHPDF(I,J),SHPDB(I,J),SHPDY(I,J),SHPDL(I,J)=SETUP HRS/DAY,FELL,ETC.
0178    C  TSHS(I)=TOTAL SETUP HRS FOR SYSTEM,UNIT I
0179    C  TSCS(I)=TOTAL SETUP COST FOR SYSTEM,UNIT I
0180    C  PASH(I)=PRODUCTION AND SETUP HRS, UNIT I
0181    C  PASC(I)=PRODUCTION AND SETUP COST, UNIT I
0182    C  CPTUU(I)=PRODUCTION AND SETUP COST PER TON
0183    42    DO 204 I=1,NUNITS
0184          NN=ZUNIT(I)
0185          DO 205 J=1,NN
0186          M=TMDF(I,J)+1.0
0187          DAYF(I,J)=M
0188          MM=TMDB(I,J)+1.0
0189          DAYB(I,J)=MM
0190          MMM=TMDY(I,J)+1.0
0191          DAYY(I,J)=MMM
0192          MMMM=TMDL(I,J)+1.0
0193          DAYL(I,J)=MMMM
0194          TSHF(I)=TSHF(I)+SHPDF(I,J)*DAYF(I,J)
0195          TSHB(I)=TSHB(I)+SHPDB(I,J)*DAYB(I,J)
0196          TSHY(I)=TSHY(I)+SHPDY(I,J)*DAYY(I,J)
0197          TSHL(I)=TSHL(I)+SHPDL(I,J)*DAYL(I,J)
0198          TSCF(I)=TSCF(I)+SHPDF(I,J)*SCPHF(I,J)*DAYF(I,J)
0199          TSCB(I)=TSCB(I)+SHPDB(I,J)*SCPHB(I,J)*DAYB(I,J)
0200          TSCY(I)=TSCY(I)+SHPDY(I,J)*SCPHY(I,J)*DAYY(I,J)
0201          TSCL(I)=TSCL(I)+SHPDL(I,J)*SCPHL(I,J)*DAYL(I,J)
0202    205   CONTINUE
0203          TSHS(I)=TSHF(I)+TSHB(I)+TSHY(I)+TSHL(I)
0204          TSCS(I)=TSCF(I)+TSCB(I)+TSCY(I)+TSCL(I)
0205          PASH(I)=TSHS(I)+T(I)
0206          PASC(I)=TSCS(I)+C(I)
0207          CPTUU(I)=PASC(I)/P(I)
0208    204   CONTINUE
0209          IF(KK.EQ.99) GO TO 800
0210          WRITE(9,55)
0211     55   FORMAT(54X,"UNIT   SUMMARY   REPORT",//)
0212          WRITE(9,47)
0213    47    FORMAT(47X,"HOURS",10X,"TONS",10X,"COST",10X,"$/TON",///)
0214     54   DO 57 I=1,NUNITS
```

```
0215          CPTU(I)=C(I)/P(I)
0216          TPHU(I)=P(I)/T(I)
0217          CPHU(I)=C(I)/T(I)
0218    C  OUTPUT UNIT DATA
0219          WRITE(9,56)I,T(I),P(I),C(I),CPTU(I)
0220     56   FORMAT(20X,'UNIT(',I2,')',2X,'PRODUCTIVE',4X,F8.2,
0221         16X,F8.2,6X,F8.2,7X,F8.2,/)
0222          WRITE(9,202)TSHS(I),TSCS(I)
0223     202  FORMAT(30X,'SETUP',9X,F8.2,20X,F8.2,/)
0224          WRITE(9,203)PASH(I),P(I),PASC(I),CPTUU(I)
0225     203  FORMAT(30X,'TOTAL',9X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,//)
0226     57   CONTINUE
0227    C
0228    C OUTPUT THE SITE TOTALS BY MATRIX
0229    C  J=INDEX ON LU(DISC FILE TO READ FROM)
0230    C CALCULATE AND OUTPUT TOTAL COST,PRODUCTION,AND TIME.
0231    C CALCULATE AND OUTPUT COST/HR,COST/TON, TONS/HR.
0232    C TCOST= THE TOTAL COST OF HARVESTING SITE ($), TTIM=
0233    C THE TOTAL TIME TO HARVEST SITE (HRS.), TPRD= THE TOTAL
0234    C PRODUCTION FROM THE SITE (TONS), CTHR= COST PER HOUR
0235    C ($/HR), TNHR= THE TONS OF PRODUCTION PER HOUR (TONS/HR)
0236    C CTTN= THE COST PER TON ($/TON)
0237    C
0238     44   TTIM=0.
0239          TCOST=0.
0240          DO 40 I=1,4
0241          TCOST=TCOST+TOTCT(I)
0242          TTIM=TTIM+TOTTM(I)
0243       40 CONTINUE
0244          TPRD=TOTPR(3)
0245          CTHR=TCOST/TTIM
0246          TNHR=TPRD/TTIM
0247          CTTN=TCOST/TPRD
0248    C  STSH=SITE TOTAL SETUP HOURS
0249    C  STSC=SITE TOTAL SETUP COSTS
0250    C  SPASH=SITE PRODUCTION AND SETUP HOURS
0251    C  SPASC=SITE PRODUCTION AND SETUP COSTS
0252    C  SPACT=SITE PRODUCTION AND SETUP COSTS/TON
0253          STSH=0.
0254          STSC=0.
0255          DO 206 I=1,NUNITS
0256          STSH=TSHS(I)+STSH
0257          STSC=TSCS(I)+STSC
0258     206  CONTINUE
0259          SPASH=STSH+TTIM
0260          SPASC=STSC+TCOST
0261          SPACT=SPASC/TPRD
0262          WRITE(9,48)TTIM,TPRD,TCOST,CTTN
0263     48   FORMAT(17X,'SITE TOTALS',2X,'PRODUCTIVE',4X,F8.2
0264         1,6X,F8.2,6X,F8.2,7X,F8.2,/)
0265          WRITE(9,207)STSH,STSC
0266     207  FORMAT(30X,'SETUP',9X,F8.2,20X,F8.2,/)
0267          WRITE(9,208)SPASH,TPRD,SPASC,SPACT
0268     208  FORMAT(30X,'TOTAL',9X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,///)
```

```
0269         WRITE(9,74)
0270         CALL TINPT(ICHAR)
0271         CALL NEWPG
0272   C
0273   C  UNIT SUMMARY - ACTIVITIES
0274   C
0275   C  TSUM=TOTAL OF ACTIVITIES FOR SUMMARY
0276   C  SUMT(K)=TOTAL FOR ACTIVITY (K)
0277   C  PEC(K)=PERCENTAGE OF THE SUMMARY ANALYSIS FOR ACTIVITY (K)
0278   C  R(I,K)=PERCENTAGE OF ACTIVITY (K) UNIT (I)
0279   C  DPT(I,K)=SUMMARY RATE CALCULATED FOR UNIT (I), ACTIVITY
0280   C  CSUM(K)=SUM OF THE COST FOR ACTIVITY (K)
0281   C  PSUM(K)=SUM OF THE PRODUCTION FOR ACTIVITY (K)
0282   C  HSUM(K)=SUM OF THE HOURS FOR ACTIVITY (K)
0283   C  WAVG(K)=WEIGHTED AVERAGE
0284         ICT = 0
0285   C
0286         DO 699 I=1,NUNITS
0287         AT(I) = 0.0
0288    699  CONTINUE
0289   C
0290         DO 700 I=1,NUNITS
0291         DO 701 J=1,ZUNIT(I)
0292         DO 702 K=1,NZ(I,J)
0293         ICT = ICT + 1
0294         AT(I) = AT(I) + AS(I,J,K)
0295    702  CONTINUE
0296    701  CONTINUE
0297    700  CONTINUE
0298   C
0299    800  WRITE(9,801)
0300    801  FORMAT(1X,'ENTER 88 IF UNIT ACTIVITY SUMMARY IS DESIRED  ',
0301        1'99 FOR NONE ----> _')
0302         READ(9,*) MM
0303         CALL NEWPG
0304         IF (MM-88) 800, 802, 62
0305    802  WRITE(9,803)
0306    803  FORMAT(46X,'UNIT ACTIVITY SUMMARY',//)
0307         WRITE(9,209)
0308    209  FORMAT(44X,'(PRODUCTIVE HOURS & COSTS)',//)
0309         WRITE(9,804)
0310    804  FORMAT(8X,'AREA',5X,'(TONS)',9X,'FELLING',12X,'BUNCHING',13X,
0311        1'YARDING',13X,'LANDING')
0312         WRITE(9,805)
0313    805  FORMAT(1X,'UNIT*',1X,'(ACRES)',3X,'VOLUME',5X,
0314        14('HOURS',6X,'COST',5X),//)
0315         DO 806 I=1,NUNITS
0316         WRITE(9,807) I,AT(I),PPRU(I),TT(I,1),CC(I,1),TT(I,2),CC(I,2),
0317        1TT(I,3),CC(I,3),TT(I,4),CC(I,4)
0318    807  FORMAT(2X,I2,1X,'*',F7.2,2X,9(F8.2,2X))
0319    806  CONTINUE
0320   C  PASHF(I),PASHB(I),PASHY(I),PASHL(I)=PROD & SETUP HRS,FELL,ETC.
0321   C  PASCF(I),PASCB(I),PASCY(I),PASCL(I)=PROD & SETUP $,FELL,ETC.
0322         WRITE(9,210)
```

```
0323   210  FORMAT(1X,////,47X,"(TOTAL HOURS & COSTS)",//)
0324        WRITE(9,804)
0325        WRITE(9,805)
0326        DO 211 I=1,NUNITS
0327        PASHF(I)=TSHF(I)+TT(I,1)
0328        PASCF(I)=TSCF(I)+CC(I,1)
0329        PASHB(I)=TSHB(I)+TT(I,2)
0330        PASCB(I)=TSCB(I)+CC(I,2)
0331        PASHY(I)=TSHY(I)+TT(I,3)
0332        PASCY(I)=TSCY(I)+CC(I,3)
0333        PASHL(I)=TSHL(I)+TT(I,4)
0334        PASCL(I)=TSCL(I)+CC(I,4)
0335        WRITE(9,807)I,AT(I),PPRU(I),PASHF(I),PASCF(I),PASHB(I),PASCB(I)
0336       1,PASHY(I),PASCY(I),PASHL(I),PASCL(I)
0337   211  CONTINUE
0338        WRITE(9,74)
0339        CALL TINPT(ICHAR)
0340        CALL NEWPG
0341   62   WRITE(9,61)
0342   61   FORMAT(40X,"ENTER THE APPROPRIATE NUMBER FOR PARAMETER SUMMARY",
0343       1///,62X,"0=NONE",/,62X,"1=HOURS",/,62X,"2=COST",/,62X,"3=$/TON",
0344       2/,62X,"4=TONS/HR",/,62X,"5=$/HR",//,
0345       362X,"---------> _")
0346        READ(9,*)M
0347        CALL NEWPG
0348        IF(M) 62,109,63
0349   63   GO TO (64,68,80,85,89),M
0350   C
0351   C  WRITE OUT HOURS TABLE
0352   C
0353    64   WRITE(9,65)
0354    65   FORMAT (56X,"SUMMARY FOR HOURS (%)")
0355         WRITE(9,212)
0356    212  FORMAT(58X,"(PRODUCTIVE HOURS)",///)
0357         WRITE(9,66)
0358    66   FORMAT(22X,"UNIT #",8X,"FELLING",10X,"BUNCHING",10X,"YARDING",
0359        110X,"LANDING",10X,"TOTAL",//)
0360         DO 67 I=1,NUNITS
0361         DO 67 K=1,4
0362         R(I,K)=(TT(I,K)/T(I))*100.
0363    67   CONTINUE
0364         DO 72 I=1,NUNITS
0365         WRITE(9,73)I,TT(I,1),R(I,1),TT(I,2),R(I,2),TT(I,3),R(I,3),TT(I,4),
0366        1R(I,4),T(I)
0367    73   FORMAT(24X,I1,2X,"#",2X,4(F8.2,1X,"(",F3.0,"%)",2X,),F8.2,
0368        11X,"(100%)")
0369    72   CONTINUE
0370         TSUM=0.
0371         DO 196 K=1,4
0372         SUMT(K)=0.
0373    196  CONTINUE
0374         DO 170 K=1,4
0375         DO 171 I=1,NUNITS
0376         SUMT(K)=SUMT(K)+TT(I,K)
```

```
0377   171  CONTINUE
0378   170  CONTINUE
0379        DO 172 K=1,4
0380        TSUM=TSUM+SUMT(K)
0381   172  CONTINUE
0382        DO 173 K=1,4
0383        PEC(K)=(SUMT(K)/TSUM)*100.
0384   173  CONTINUE
0385        WRITE(9,174)SUMT(1),PEC(1),SUMT(2),PEC(2),SUMT(3),PEC(3),
0386       1SUMT(4),PEC(4)
0387   174  FORMAT(20X,'TOTAL',2X,'*',2X,4(F8.2,1X,'(',F3.0,'%)',2X))
0388  C  Q(I),R(I),V(I),E(I)=SETUP $ AS % OF TOTAL,FELL,ETC.
0389        WRITE(9,213)
0390   213  FORMAT(1X,/////,51X,'NON-PRODUCTIVE(SETUP) HOURS',///)
0391        WRITE(9,66)
0392        F=0.
0393        G=0.
0394        FX=0.
0395        FG=0.
0396        DO 214 I=1,NUNITS
0397        Q(I)=(TSHF(I)/TSHS(I))*100.
0398        RR(I)=(TSHB(I)/TSHS(I))*100.
0399        V(I)=(TSHY(I)/TSHS(I))*100.
0400        E(I)=(TSHL(I)/TSHS(I))*100.
0401        F=TSHF(I)+F
0402        G=TSHB(I)+G
0403        FX=TSHY(I)+FX
0404        FG=TSHL(I)+FG
0405   214  CONTINUE
0406        X1=(F/STSH)*100.
0407        X2=(G/STSH)*100.
0408        X3=(FX/STSH)*100.
0409        X4=(FG/STSH)*100.
0410        DO 215 I=1,NUNITS
0411        WRITE(9,73)I,TSHF(I),Q(I),TSHB(I),RR(I),TSHY(I),V(I)
0412       1,TSHL(I),F(I),TSHS(I)
0413   215  CONTINUE
0414        WRITE(9,174)F,X1,G,X2,FX,X3,FG,X4
0415  C  A1,A2,A3,A4=% FOR COMBINED TOTALS
0416  C  COMT1,COMT2,...=COMBINED TOTAL PROD & SETUP HRS,FELL,ETC.
0417        COMT1=SUMT(1)+F
0418        COMT2=SUMT(2)+G
0419        COMT3=SUMT(3)+FX
0420        COMT4=SUMT(4)+FG
0421        GTOT=COMT1+COMT2+COMT3+COMT4
0422        A1=(COMT1/GTOT)*100.
0423        A2=(COMT2/GTOT)*100.
0424        A3=(COMT3/GTOT)*100.
0425        A4=(COMT4/GTOT)*100.
0426        WRITE(9,216)COMT1,A1,COMT2,A2,COMT3,A3,COMT4,A4
0427   216  FORMAT(1X,///,11X,'COMBINED TOTAL',2X,'*',2X
0428       1,4(F8.2,1X,'(',F3.0,'%)',2X))
0429   78   WRITE(9,74)
0430   74   FORMAT(1X,////,'PRESS RETURN KEY TO CONTINUE')
```

```
0431          CALL TINPT (ICHAR)
0432          CALL NEWPG
0433          GO TO 62
0434    C
0435    C  WRITE OUT COST TABLE
0436    C
0437     68   WRITE(9,75)
0438     75   FORMAT(56X,'SUMMARY FOR COSTS (%)',//)
0439          WRITE(9,217)
0440     217  FORMAT(57X,'(PRODUCTIVE COSTS)',///)
0441          WRITE (9,66)
0442          DO 76 I=1,NUNITS
0443          DO 76 K=1,4
0444          R(I,K)=(CC(I,K)/C(I))*100.
0445     76   CONTINUE
0446          DO 77 I=1,NUNITS
0447          WRITE(9,73)I,CC(I,1),R(I,1),CC(I,2),R(I,2),CC(I,3),R(I,3),CC(I,4),
0448         1R(I,4),C(I)
0449     77   CONTINUE
0450          TSUM=0.
0451          DO 180 K=1,4
0452          SUMT(K)=0.
0453     180  CONTINUE
0454          TSUM=0.
0455          DO 175 K=1,4
0456          DO 176 I=1,NUNITS
0457          SUMT(K)=SUMT(K)+CC(I,K)
0458     176  CONTINUE
0459     175  CONTINUE
0460          DO 177 K=1,4
0461          TSUM=TSUM+SUMT(K)
0462     177  CONTINUE
0463          DO 178 K=1,4
0464          PEC(K)=(SUMT(K)/TSUM)*100.
0465     178  CONTINUE
0466          WRITE(9,174)SUMT(1),PEC(1),SUMT(2),PEC(2),SUMT(3),PEC(3),
0467         1SUMT(4),PEC(4)
0468    C  FTT,BTT,YTT,LTT=TOTAL OF PROD & SETUP $,FELL,BNCN,ETC.
0469    C  PF(I),PB(I),PY(I),PL(I)=% OF TOTAL PROD & SETUP $,FELL,...
0470    C  TFT,TBT,TYT,TLT=% OF SITE TOTAL PROD & SETUP $,FELL,...
0471          WRITE(9,218)
0472     218  FORMAT(1X,/////,50X,'(TOTAL PRODUCTIVE & SETUP COSTS)',///)
0473          WRITE(9,66)
0474          FTT=0.
0475          BTT=0.
0476          YTT=0.
0477          LTT=0.
0478          DO 220 I=1,NUNITS
0479          FTT=FTT+PASCF(I)
0480          BTT=BTT+PASCB(I)
0481          YTT=YTT+PASCY(I)
0482          LTT=LTT+PASCL(I)
0483          PF(I)=(PASCF(I)/PASC(I))*100.
0484          PB(I)=(PASCB(I)/PASC(I))*100.
```

```
0485          PY(I)=(PASCY(I)/PASC(I))*100.
0486          PL(I)=(PASCL(I)/PASC(I))*100.
0487    220   CONTINUE
0488          TBT=(BTT/SPASC)*100.
0489          TYT=(YTT/SPASC)*100.
0490          TLT=(LTT/SPASC)*100.
0491          TFT=(FTT/SPASC)*100.
0492          DO 221 I=1,NUNITS
0493          WRITE(9,73)I,PASCF(I),PF(I),PASCB(I),PB(I)
0494         1,PASCY(I),PY(I),PASCL(I),PL(I),PASC(I)
0495    221   CONTINUE
0496          WRITE(9,174)FTT,TFT,BTT,TBT,YTT,TYT,LTT,TLT
0497          GO TO 78
0498    C
0499    C  WRITE OUT $/TON TABLE
0500    C
0501     80   WRITE(9,81)
0502     81   FORMAT(56X,'SUMMARY FOR $/TON (%)')
0503          WRITE(9,222)
0504     222  FORMAT(56X,'(PRODUCTIVE COSTS ONLY)',//)
0505          WRITE(9,66)
0506          DO 82 I=1,NUNITS
0507          DO 82 K=1,4
0508          DPT(I,K)=CC(I,K)/PP(I,K)
0509          R(I,K)=(DPT(I,K)/CPTU(I))*100.
0510     82   CONTINUE
0511          DO 84 I=1,NUNITS
0512          WRITE(9,73)1,DPT(I,1),R(I,1),DPT(I,2),R(I,2),DPT(I,3),R(I,3),
0513         1DPT(I,4),R(I,4),CPTU(I)
0514     84   CONTINUE
0515          TSUM=0.
0516          DO 181 K=1,4
0517          CSUM(K)=0.
0518          PSUM(K)=0.
0519     181  CONTINUE
0520          DO 182 K=1,4
0521          DO 183 I=1,NUNITS
0522          CSUM(K)=CSUM(K)+CC(I,K)
0523          PSUM(K)=PSUM(K)+P(I)
0524     183  CONTINUE
0525     182  CONTINUE
0526          DO 184 K=1,4
0527          WAVG(K)=CSUM(K)/PSUM(K)
0528     184  CONTINUE
0529          DO 185 K=1,4
0530          TSUM=TSUM+WAVG(K)
0531     185  CONTINUE
0532          DO 900 K=1,4
0533          PEC(K)=(WAVG(K)/TSUM)*100.
0534     900  CONTINUE
0535     999  FORMAT(15X,'WGHT. AVG.',2X,'*',2X,4(F8.2,1X,'(',F3.0,'%)',2X))
0536          WRITE(9,999)WAVG(1),PEC(1),WAVG(2),PEC(2),WAVG(3),PEC(3),
0537         1WAVG(4),PEC(4)
0538          GO TO 78
```

```
0539  C
0540  C  WRITE OUT TONS/HR TABLE
0541  C
0542   85   WRITE(9,86)
0543   86   FORMAT(56X,'SUMMARY FOR TONS/HR (%)')
0544        WRITE(9,223)
0545   223  FORMAT(56X,'(PRODUCTIVE HOURS ONLY)',//)
0546        WRITE(9,66)
0547        DO 87 I=1,NUNITS
0548        DO 87 K=1,4
0549        DPT(I,K)=PP(I,K)/TT(I,K)
0550        TOT(I)=TOT(I)+DPT(I,K)
0551   87   CONTINUE
0552        DO 200 I=1,NUNITS
0553        DO 200 K=1,4
0554        R(I,K)=(DPT(I,K)/TOT(I))*100.
0555   200  CONTINUE
0556        DO 88 I=1,NUNITS
0557        WRITE(9,73)I,DPT(I,1),R(I,1),DPT(I,2),R(I,2),DPT(I,3),R(I,3),
0558       1DPT(I,4),R(I,4),TOT(I)
0559   88   CONTINUE
0560        TSUM=0.
0561        DO 186 K=1,4
0562        HSUM(K)=0.
0563   186  CONTINUE
0564        DO 187 K=1,4
0565        DO 188 I=1,NUNITS
0566        HSUM(K)=HSUM(K)+TT(I,K)
0567   188  CONTINUE
0568   187  CONTINUE
0569        DO 189 K=1,4
0570        WAVG(K)=PSUM(K)/HSUM(K)
0571   189  CONTINUE
0572        DO 190 K=1,4
0573        TSUM=TSUM+WAVG(K)
0574   190  CONTINUE
0575        DO 901 K=1,4
0576        PEC(K)=(WAVG(K)/TSUM)*100.
0577   901  CONTINUE
0578        WRITE(9,999)WAVG(1),PEC(1),WAVG(2),PEC(2),WAVG(3),PEC(3),
0579       1WAVG(4),PEC(4)
0580        GO TO 78
0581  C
0582  C  WRITE OUT $/HR TABLE
0583  C
0584   89   WRITE(9,90)
0585   90   FORMAT(56X,'SUMMARY FOR $/HR (%)')
0586        WRITE(9,224)
0587   224  FORMAT(49X,'(PRODUCTIVE HRS & COSTS ONLY)',//)
0588        WRITE(9,66)
0589        DO 91 I=1,NUNITS
0590        DO 91 K=1,4
0591        DPT(I,K)=CC(I,K)/TT(I,K)
0592        TOT(I)=TOT(I)+DPT(I,K)
```

```
0593   91    CONTINUE
0594         DO 201 I=1,NUNITS
0595         DO 201 K=1,4
0596         R(I,K)=(DPT(I,K)/TOT(I))*100.
0597   201   CONTINUE
0598         DO 92 I=1,NUNITS
0599         WRITE(9,73)I,DPT(I,1),R(I,1),DPT(I,2),R(I,2),DPT(I,3),R(I,3),
0600        1DPT(I,4),R(I,4),TOT(I)
0601   92    CONTINUE
0602         TSUM=0.
0603         DO 191 K=1,4
0604         WAVG(K)=CSUM(K)/HSUM(K)
0605   191   CONTINUE
0606         DO 192 K=1,4
0607         TSUM=TSUM+WAVG(K)
0608   192   CONTINUE
0609         DO 193 K=1,4
0610         PEC(K)=(WAVG(K)/TSUM)*100.
0611   193   CONTINUE
0612         WRITE(9,999)WAVG(1),PEC(1),WAVG(2),PEC(2),WAVG(3),PEC(3),
0613        1WAVG(4),PEC(4)
0614         GO TO 78
0615   109   CONTINUE
0616         WRITE(9,132)
0617   132   FORMAT(58X,'FELLING INPUTS')
0618         DO 133 I=1,NUNITS
0619         WRITE(9,134)I
0620   134   FORMAT(///,58X,'LOGGING UNIT',1X,'(',I1,')',///)
0621         WRITE(9,135)
0622   135   FORMAT(11X,'CYCLE TIME (MIN)',2X,'CYCLES',4X,'WORK HR/DAY',2X,
0623        1'PROD COST/HR',3X,'STUP HR/DAY',2X,'STUP COST/HR',2X,
0624        2'TREE WEIGHT (LBS)',//)
0625         N=ZUNIT(I)
0626         DO 136 J=1,N
0627         WRITE(9,137)J,FFLLTM(I,J),FELLNC(I,J),HPWDF(I,J),FELLCT(I,J),
0628        1SHPDF(I,J),SCPHF(I,J),FELLPR(I,J)
0629   137   FORMAT(1X,'ZONE(',I1,')',6X,3(F8.2,5X),2(F8.2,6X),F8.2,8X,F8.2)
0630   136   CONTINUE
0631   133   CONTINUE
0632         WRITE(9,139)
0633   139   FORMAT(1X,///,'PRESS RETURN KEY TO CONTINUE')
0634         CALL TINPT(ICHAR)
0635         CALL NEWPG
0636         WRITE(9,138)
0637   138   FORMAT(58X,'BUNCHING INPUTS')
0638         DO 150 I=1,NUNITS
0639         WRITE(9,134)I
0640         WRITE(9,151)
0641   151   FORMAT(14X,'CYCLE TIME (MIN)',2X,'WORK HR/DAY',2X,'PROD COST/HR',
0642        12X,'STUP HR/DAY',2X,'STUP COST/HR',2X,'LOAD WEIGHT (LBS)',//)
0643         N=ZUNIT(I)
0644         DO 152 J=1,N
0645         WRITE(9,153)J,RUNTM(I,J),HPWDB(I,J),BUNCT(I,J),SHPDB(I,J),
0646        1SCPHB(I,J),TLOAD(I,J)
```

```
0647   153  FORMAT(1X,'ZONE(',I1,')',10X,F8.2,8X,F8.2,5X,F8.2,6X,
0648       1F8.2,5X,F8.2,7X,F8.2)
0649   152  CONTINUE
0650   150  CONTINUE
0651        WRITE(9,139)
0652        CALL TINPT(ICHAR)
0653        CALL NEWPG
0654        WRITE(9,155)
0655   155  FORMAT(58X,'YARDING INPUTS')
0656        DO 154 I=1,NUNITS
0657        WRITE(9,134)I
0658        WRITE(9,156)
0659   156  FORMAT(34X,'CYCLE TIME (MIN)',2X,'WORK HR/DAY',2X,'PROD COST/HR',
0660       12X,'STUP HR/DAY',2X,'STUP COST/HR',//)
0661        N=ZUNIT(I)
0662        DO 157 J=1,N
0663        WRITE(9,158)J,YARDTM(I,J),HPWDY(I,J),YARDCT(I,J),SHPDY(I,J),
0664       1SCPHY(I,J)
0665   158  FORMAT(25X,'ZONE(',I1,')',6X,F8.2,8X,F8.2,5X,F8.2,
0666       16X,F8.2,5X,F8.2)
0667   157  CONTINUE
0668   154  CONTINUE
0669        WRITE(9,139)
0670        CALL TINPT(ICHAR)
0671        CALL NEWPG
0672        WRITE(9,160)
0673   160  FORMAT(58X,'LANDING INPUTS')
0674        DO 159 I=1,NUNITS
0675        WRITE(9,134)I
0676        WRITE(9,161)
0677   161  FORMAT(51X,'PROD COST/HR',2X,'STUP HR/DAY',2X,'STUP COST/HR',//)
0678        N=ZUNIT(I)
0679        DO 162 J=1,N
0680        WRITE(9,163)J,LANDCT(I,J),SHPDL(I,J),SCPHL(I,J)
0681   163  FORMAT(42X,'ZONE(',I1,')',5X,F8.2,5X,F8.2,6X,F8.2)
0682   162  CONTINUE
0683   159  CONTINUE
0684        CALL SPOLU(LU(1),FELL1,1,2,25)
0685        CALL SPOLU(LU(2),BNCH1,1,2,25)
0686        CALL SPOLU(LU(3),YARD1,1,2,25)
0687        CALL SPOLU(LU(4),LAND1,1,2,25)
0688        WRITE(9,120)
0689    120 FORMAT(1X,'*STOP*')
0690        END
0691   C
0692   C SUBROUTINE SWAP
0693   C
0694        SUBROUTINE SWAP (PROGRM)
0695   C
0696   C MODULE TO PERFORM PROGRAM SWAPS
0697   C
0698        COMMON ITEK(45)
0699        COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
0700       1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
```

```
0701        2PHPDY(5,5),FELLTM(5,5),PR(5,5),DA
0702  C
0703         EQUIVALENCE (COMMON(1),ITEK(1))
0704  C
0705         INTEGER UNITS,DA
0706         INTEGER PROGRM(3), RP, OF, COMMON(1), IPRM(5),ZUNIT
0707  C
0708         DATA RP / 1 /, OF / 2 /
0709  C
0710  C SWAP
0711  C
0712         IBUFL=390
0713         IF(PROGRM(1).EQ.2HSO)IBUFL=57
0714         IF (PROGRM(1).EQ.2HFE) IBUFL = 491
0715         CALL IDSEG (PROGRM, RP)
0716         CALL NEWPG
0717         IPRM(1)=9
0718         CALL EXEC (9,PROGRM, IPRM(1),IPRM(2),IPRM(3),IPRM(4),IPRM(5),
0719        &  COMMON, IBUFL)
0720         CALL EXEC (14,1,COMMON,IBUFL)
0721         CALL IDSEG (PROGRM, OF)
0722  C
0723         RETURN
0724  C
0725         END
0726         END$
```

```
&HSTDS T=00004 IS ON CR00025 USING 00027 BLKS R=0014

0001  FTN4
0002        SUBROUTINE DIGBT (AU,AZ,AS,II,NZ)
0003        COMMON DUMMY(45),UNITS,ZUNITS(5),ISTA(5),I1,I2,I3
0004        INTEGER UNITS,ZUNITS,ISTA,ITMP(50)
0005        INTEGER NZTMP(5,5),II(50),NZ(5,5),NUNITS,NZUN(5)
0006        REAL AREA(50)
0007        INTEGER BUF(13),RBUF(181)
0008        REAL AU(5),AZ(5,5),AS(5,5,20)
0009        INTEGER DUMMY
0010        EQUIVALENCE (BUF(1),DUMMY(44))
0011        EQUIVALENCE (RBUF(1),AREA(1)),
0012       &                (RBUF(101),ITMP(1)),
0013       &                 (RBUF(151),NZTMP(1,1)),
0014       &                  (RBUF(176),NUNITS),
0015       &                   (RBUF(177),NZUN(1))
0016        INTEGER DIGBX(3)
0017        DATA DIGBX/2HDI,2HGB,2HX /
0018        CALL IDSEG(DIGBX,1)
0019        CALL EXEC(9,DIGBX,9,I2,I3,I4,I5,BUF,13)
0020        CALL EXEC(14,1,RBUF,181)
0021        ICT = 0
0022        KN = 0
0023        DO 70 I=1,5
0024        DO 75 J=1,5
0025        NZ(I,J) = NZTMP(I,J)
0026   75   CONTINUE
0027   70   CONTINUE
0028        DO 80 I=1,50
0029        II(I) = ITMP(I)
0030   80   CONTINUE
0031        UNITS=NUNITS
0032        DO 90 I=1,5
0033        ZUNITS(I) = NZUN(I)
0034   90   CONTINUE
0035        DO 100 K=1,UNITS
0036        ICT = ICT + 1
0037        AU(K) = AREA(ICT)
0038        IF(ZUNITS(K).EQ.1)GO TO 95
0039        DO 110 J=1,ZUNITS(K)
0040        ICT = ICT + 1
0041        AZ(K,J) = AREA(ICT)
0042        GO TO 96
0043   95   AZ(K,1) = AU(K)
0044        J=1
0045   96   DO 120 L=1,NZ(K,J)
0046        ICT = ICT + 1
0047        KN = KN + 1
0048        AS(K,J,L) = AREA(ICT)
0049   120  CONTINUE
0050        IF(ZUNITS(K).EQ.1) GO TO 100
0051   110  CONTINUE
0052   100  CONTINUE
```

```
0053          CALL IDSEG(DIGBX,2)
0054          RETURN
0055          END
0056    C  ********
0057    C  ********
0058          SUBROUTINE SORT(FNCDUM,PRDUM,FTMDUM,DA,AUD,AZD,ASD,IID,NZD)
0059          COMMON ITEK(45)
0060          COMMON UNITS,ZUNIT(5),ISTA(5)
0061          INTEGER IID(50),NZD(5,5)
0062          INTEGER COMMON(59),ZUNIT,SORTX(3),II(50),NZ(5,5)
0063          INTEGER UNITS,DA,DDA
0064          INTEGER COMBL
0065          REAL AUD(5),AZD(5,5),ASD(5,5,20)
0066          REAL AU(5),AZ(5,5),AS(5,5,20)
0067          REAL FNCDUM(5,5),PRDUM(5,5),FTMDUM(5,5)
0068          REAL FELLNC(5,5),PR(5,5),FELLTM(5,5)
0069          INTEGER BUF(1285),TRKSAV
0070          EQUIVALENCE (BUF(1),FELLNC(1,1)),(BUF(51),PR(1,1)),
0071         &(BUF(101),FELLTM(1,1)),(BUF(151),AU(1)),
0072         &(BUF(161),AZ(1,1)),(BUF(211),AS(1,1,1)),
0073         &(BUF(1211),II(1)),(BUF(1261),NZ(1,1))
0074          EQUIVALENCE (COMMON(1),ITEK(1)),
0075         &(COMMON(57),ITRK),(COMMON(58),ISECT),(COMMON(59),DDA)
0076          DATA SORTX/2HSO,2HRT,2HX /
0077          COMBL=59
0078          CALL IDSEG(SORTX,1)
0079          CALL EXEC(15,1,ITRK,IDISC,ISECT)
0080          TRKSAV = ITRK
0081          ISECT = 0
0082          CALL EXEC(9,SORTX,COMBL,COMBL,I3,I4,I5,COMMON,COMBL)
0083          COMBL=59
0084          CALL EXEC(14,1,COMMON,COMBL)
0085          DA=DDA
0086          IBUFL=1285
0087          ITRK = TRKSAV
0088          ISECT = 0
0089          CALL EXEC(1,66,BUF,IBUFL,ITRK,ISECT)
0090    C
0091          DO 30 I=1,5
0092    C
0093             AUD(I) = AU(I)
0094    C
0095             DO 40 J=1,5
0096    C
0097             FNCDUM(I,J) = FELLNC(I,J)
0098             FTMDUM(I,J) = FELLTM(I,J)
0099             PRDUM(I,J) = PR(I,J)
0100             AZD(I,J) = AZ(I,J)
0101             NZD(I,J) = NZ(I,J)
0102    C
0103             DO 50 K=1,20
0104    C
0105                ASD(I,J,K) = AS(I,J,K)
0106    C
```

```
0107   50          CONTINUE
0108   40       CONTINUE
0109   30    CONTINUE
0110         DO 60 I=1,50
0111            IID(I) = II(I)
0112   60    CONTINUE
0113         CALL EXEC(16,1,ITRK,IDISC)
0114         CALL IDSEG(SORTX,2)
0115         RETURN
0116         END
0117   C  *******
0118   C  *******
0119         SUBROUTINE A2DRT(AREA,ITOT,FILNM)
0120         DIMENSION AREA(50),BUFF(50)
0121         INTEGER FILNM(8),IBUF(9),BUFFER(200),A2DBX(3)
0122         EQUIVALENCE (BUFFER(1),BUFF(1)),(BUFFER(101),IBUF(1))
0123         DATA A2DBX/2HA2, 2HDB, 2HX  /
0124         IBUF(1)=ITOT
0125         DO 20 I=1,8
0126   20    IBUF(I+1)=FILNM(I)
0127         DO 25 I=1,50
0128   25    BUFF(I)=AREA(I)
0129         IBUFL=109
0130         CALL IDSEG(A2DBX,1)
0131         CALL EXEC(9,A2DBX,0,0,0,0,0,BUFFER,IBUFL)
0132         CALL EXEC(14,1,BUFFER,IBUFL)
0133         CALL IDSEG(A2DBX,2)
0134         ITOT=IBUF(1)
0135         DO 30 I=1,50
0136   30    AREA(I)=BUFF(I)
0137         RETURN
0138         END
0139   C  *******
0140   C*************** SUBROUTINE PUTCM ********************
0141   C
0142         SUBROUTINE PUTCM(ICOM,II)
0143   C
0144   $INSERT /SWPCM::25
0145   C
0146         DIMENSION ICOM(1)
0147   C
0148         I=II
0149   C$$
0150         WRITE(6,9000) I,ISECT(I),ICL(I)
0151    9000 FORMAT(' IN PUTCM I,ISECT,ICL =',3I5)
0152         CALL EXEC(2,66,ICOM,ICL(I),ISTRK(I),ISECT(I))
0153   C
0154         RETURN
0155         END
0156   C*************** SUBROUTINE GETCM ********************
0157   C
0158         SUBROUTINE GETCM(ICOM,II)
0159   C
0160   $INSERT /SWPCM::25
```

```
0161  C
0162        DIMENSION ICOM(1)
0163  C
0164        I=II
0165  C$$
0166        WRITE(6,9000) I,ISECT(I),ISTRK(I),ICL(I)
0167   9000 FORMAT(' IN GFTCM I,ISECT,ISTRK,ICL=',4I5)
0168        CALL EXEC(1,66,ICOM,ICL(I),ISTRK(I),ISECT(I))
0169  C
0170        RETURN
0171        END
0172  C*************** SUBROUTINE DMPCM ***************************
0173        SUBROUTINE DMPCM(ICOM,I,NAM)
0174        REAL NAM
0175        DIMENSION ICOM(1),NAM(2)
0176  C
0177  $INSERT /SWPCM::25
0178  C
0179        NN=ICL(I)
0180        WRITE(6,9000) (NAM(K),K=1,2),I, (ICOM(J),J=1,NN)
0181   9000 FORMAT(' IN ',2A2,' ICOM #',I5,' =  ',/10(10I10/))
0182        RETURN
0183        END
0184  C ** ROUTINE TO STOP OUTPUT AND CLEARS SCREEN WHEN USER ENTERS A RESPONSE
0185  C
0186  C L IS L.U. WHERE MESSAGE IS TO BE SENT AND WHERE RESPONSE IS TO BE READ
0187  C
0188  C ** PUT IN 5/13/81    D.B. **
0189  C
0190        SUBROUTINE PAUSE(L)
0191        INTEGER ANSWER,L
0192        COMMON ITEK(45)
0193  C ** IF L IS NEG. THEN JUST CLEAR THE SCREEN **
0194        IF (L.LT.0) GO TO 05
0195        WRITE(L,10)
0196  10    FORMAT(' ENTER A SPACE FOLLOWED BY A CARRIAGE',
0197       1        ' RETURN TO CONTINUE')
0198        READ(L,20) ANSWER
0199  20    FORMAT(A2)
0200  05    CALL INITT(60)
0201        CALL CHSIZ(4)
0202        CALL ERASE
0203        RETURN
0204        END
0205  C*************** BLOCK DATA SUBROUTINE ***************************
0206  C
0207        BLOCK DATA
0208  C
0209  $INSERT      /SWPCM::25
0210  C
0211        END
0212        END$
```

```
&A2DBX T=00004 IS ON CR00025 USING 00004 BLKS R=0012

0001  FTN
0002        PROGRAM A2DBX
0003        REAL AREA(50),BUFF(50)
0004        INTEGER IPRM(5),FILNM(8),BUFFER(109)
0005        EQUIVALENCE (BUFFER(1),BUFF(1)),
0006       2 (BUFFER(101),ITOT),
0007       3 (BUFFER(102),FILNM(1))
0008        CALL RMPAR(IPRM)
0009        IBUFL = 109
0010        CALL EXEC(14,1,BUFFER,IBUFL)
0011        CALL A2DBT(AREA,ITOT,FILNM)
0012        DO 13 I=1,50
0013        BUFF(I) = AREA(I)
0014   13   CONTINUE
0015        CALL EXEC(14,2,BUFFER,IBUFL)
0016        END
```

```
&A2DBT T=00004 IS ON CR00025 USING 00026 BLKS R=0027

0001  FTN
0002  C-------------------------------------------------------------------
0003  C
0004  C            GEOSCAN planar area calculation program   (A2DBT)
0005  C
0006  C            Written by M. Pexton                      July, 1981
0007  C            MODIFIED BY B. TAYLOR                    MARCH, 1982
0008  C       This program computes the planar area (2-dimensional) of each
0009  C    area in a GEOSCAN area type data file.
0010  C       The input data file name is asked for by A2DBT. The user is also
0011  C    asked whether the computed area for each area record should be out-
0012  C    put, and if so, the area values are typed at the users terminal.
0013  C    A2DBT always prints the total area computed for all area records in
0014  C    the data file before terminating.
0015  C
0016  C       The following polygon area formula is used to compute areas:
0017  C
0018  C          AREA = (X1*Y2 + X2*Y3 + ... Xn-1*Yn + Xn*Y1
0019  C                - Y1*X2 - Y2*X3 - ... Yn-1*Xn - Yn*X1)/2
0020  C
0021  C
0022  C       A2DBT is written in FORTRAN IV and is implemented on the HP
0023  C    9640A computer with the HP RTE-III operating system.
0024  C
0025  C          Note: In this implementation, the HP RTE-III operating
0026  C                system command BR,A2DBT causes A2DBT to abort.
0027  C
0028  C       This program refers to the GSFIO and F1FIO libraries for file
0029  C    I/O, the TRMIO library for terminal I/O, and the STRNG library
0030  C    for character string manipulations.
0031  C
0032  C       The following global variables are defined by A2DBT:
0033  C
0034        BLOCK DATA
0035  $INSERT /A2DBT::25     /* A2DBT program variables */
0036        END
0037  C.F
0038  C-------------------------------------------------------------------
0039  C
0040  C            GEOSCAN planar area calculation program
0041  C
0042        SUBROUTINE A2DBT(AREA,ITOT,FILNM)
0043  C
0044  C - Global variables:
0045  $INSERT /A2DBT::25     /* A2DBT program variables */
0046  C
0047  C - Global constants:
0048  $INSERT /GFTYP::25     /* GEOSCAN data file types */
0049  $INSERT /F1FIO::25     /* File I/O (version 1.0) constants */
0050  $INSERT /TRMIO::25     /* Terminal I/O constants & variables */
0051  $INSERT /STRNG::25     /* String status constants */
0052  C
```

```
C - Functions:
      INTEGER  FOPEN, RFHED, RSHED, RFDAT
      INTEGER  TLEN,FILNM(8)
C
C - Local variables:
      INTEGER  IPARM(5)     /* for HP system
      INTEGER  FSTATS
      INTEGER  IFNAME(8), MAXIFL, IFLEN     /* Input file name
      INTEGER  PNAME(8), MAXPNL, PNLEN      /* Program name
      INTEGER  GFTYPE     /* GEOSCAN file type
      INTEGER  TITLE(40), MAXTL, TITLEN     /* File title
      INTEGER  NPTS, NCOL, MAXCOL     /* No. of points, columns/record
      INTEGER  ACOUNT     /* Area count
      LOGICAL  EVERYA     /* If TRUE, every area value printed
      REAL  DMIN(2), DMAX(2)     /* Data min/max coordinates
      REAL  XMIN, XMAX, YMIN, YMAX   /* Data min/max values
      REAL  XX, YY                /* Input data
      DOUBLE PRECISION  TQUAN,    /* Quantities for area calculation
     &                  QUAN1, QUAN2
      REAL  AREA,  TAREA          /* Area and total area computed
      DIMENSION AREA(50)
C
      DATA  MAXCOL/2/, MAXIFL/14/, MAXPNL/16/, MAXTL/80/
C
C * Begin A2DBT
C
      OK = 1
      ERR = -1
      READIT = 1
      WRITIT = 2
      TRMLU = 9
C
      INFIL = 0
C
      DO 66 I=1,8
      IFNAME(I)=FILNM(I)
 66   CONTINUE
      IFLEN = TLEN (IFNAME,1,8,' ')
C
      IF (FOPEN (INFIL,IFNAME,1,IFLEN,OLDFIL,READIT) .NE. OK)
     &   CALL ABEND
C
      IF (RFHED (INFIL,GFTYPE,PNAME,MAXPNL,PNLEN,TITLE,MAXTL,TITLEN,
     &           XMIN,YMIN,XMAX,YMAX) .NE. OK) CALL ABEND
      IF (.NOT. (GFTYPE .EQ. AREAF) ) GOTO 8
C
C        CALL SAY (NOTHNG,0)
         TAREA = 0.0
         ACOUNT = 0
C
      DO 100 I=1,ITOT
      AREA(I)=0.
 100  CONTINUE
 2    DO 101 J=1,ITOT
```

```
0107            QUAN1 = 0.0
0108            QUAN2 = 0.0
0109            TQUAN = 0.0
0110            NRECRD = 1
0111            FSTATS = RSHED (INFIL,GFTYPE,NPTS,MAXCOL,NCOL,DMIN,DMAX)
0112            IF (FSTATS .EQ. ERR) CALL ABEND
0113            IF (.NOT. (FSTATS .EQ. OK) ) GOTO 4
0114   21          IF (GFTYPE .NE. AREAF) GOTO 2     /* Repeat
0115   C
0116               IF (RFDAT (INFIL,XX,YY) .NE. OK) CALL ABEND
0117               ACOUNT = ACOUNT + 1
0118               PREVX = XX
0119               PREVY = YY
0120               NRECRD = NRECRD + 1
0121   C
0122               DO 25 I=NRECRD,NPTS
0123                  IF (IFBRK(A2DBT)) 23,24      /* HP routine
0124   23                CALL ERROR (USRERR,'User BREAK recieved.',20,0,NULL)
0125                     CALL ABEND
0126   C
0127   24             IF (RFDAT (INFIL,XX,YY) .NE. OK) CALL ABEND
0128               QUAN1 = DBLE(PREVX) * DBLE(YY)
0129               QUAN2 = DBLE(XX) * DBLE(PREVY)
0130               TQUAN = TQUAN + (QUAN1 - QUAN2)
0131               PREVX = XX
0132               PREVY = YY
0133   25          CONTINUE
0134   C
0135            AREA(J) = ABS( SNGL( TQUAN/2.0 ) )
0136   C
0137    101  CONTINUE
0138    4       CALL FCLOS (INFIL,SAVE)
0139   C
0140            RETURN
0141   C
0142    8     CALL ERROR (USRERR,'Wrong type data file. ',22,
0143         &                    'Should be AREA type.',20)
0144          CALL ABEND
0145          END
0146   END$
```

```
&DIGBX T=00004 IS ON CR00025 USING 00008 BLKS R=0009

0001  FTN4
0002        PROGRAM DIGBX
0003  $INSERT     /TKTRX::25
0004        COMMON IIX,IIY
0005        INTEGER BUF(13)
0006        EQUIVALENCE                (BUF(3),UNITS),(BUF(4),ZUNIT(1)),
0007       &(BUF(9),ISTA(1))
0008  $INSERT     /DIGBT::25
0009  $INSERT     /DGVAR::25
0010  $INSERT     /F2VAR::25
0011  $INSERT     /PSTAT::25
0012  $INSERT     /FSTAT::25
0013  $INSERT     /DSTAT::25
0014  $INSERT     /DEVCE::25
0015  $INSERT     /DCODS::25
0016  $INSERT     /PROGW::25
0017  $INSERT     /TRMIO::25
0018  $INSERT     /STRNG::25
0019  $INSERT     /GFTYP::25
0020        REAL AREA(50)
0021        INTEGER II(50),NZ(5,5)
0022        INTEGER TII(50),TAS(5,5),TZUN(5),NUNITS
0023        REAL TAREA(50)
0024        INTEGER RBUF(181)
0025        EQUIVALENCE (RBUF(1),TAREA(1)),(RBUF(101),TII(1)),
0026       &(RBUF(151),TAS(1)),(RBUF(176),NUNITS),(RBUF(177),TZUN(1))
0027        INTEGER UNITS,ZUNIT(5),ISTA(5)
0028        INTEGER IPRM(5)
0029  C
0030        CALL RMPAR(IPRM)
0031        CALL EXEC (14,1,BUF,13)
0032        TRMLU = 9
0033        CALL DIGBT(UNITS,ZUNIT,ISTA,AREA,II,NZ)
0034        DO 30 I=1,50
0035        TAREA(I) = AREA(I)
0036   30   CONTINUE
0037        DO 45 I=1,50
0038        TII(I) = II(I)
0039   45   CONTINUE
0040        NUNITS = UNITS
0041        DO 40 I=1,5
0042        TZUN(I) = ZUNIT(I)
0043   40   CONTINUE
0044        DO 50 I=1,5
0045        DO 50 K=1,5
0046        TAS(I,K)=NZ(I,K)
0047   50   CONTINUE
0048        CALL EXEC(14,2,RBUF,181)
0049        END
0050        END$
```

```
&DIGBT T=00004 IS ON CR00025 USING 00070 BLKS R=0008

0001  FTN
0002  C-------------------------------------------------------------------------
0003  C
0004  C    lines, or areas may be input with DIGBT.
0005  C
0006  C       Procedure for using DIGBT is to place a map on the digitizing
0007  C    tablet, answer the program's questions at the terminal, digitize
0008  C    the corners of the chart with the tablet's cursor (cross hairs or
0009  C    pen), and then input the points with the cursor. The points input
0010  C    are drawn on the terminal screen as they are input. A keyboard
0011  C    character (followed by a return in this implementation) input
0012  C    at the terminal suspends the data input at the tablet and the
0013  C    program control options can be input at the terminal. Typing
0014  C    HELP will list the available user options. The option commands
0015  C    can usually be shortened to one character (the first character)
0016  C    commands.
0017  C
0018  C       During the initial question and answer phase, DIGBT asks
0019  C    for the true values of the map's lower left corner and the upper
0020  C    right corner. When a point is input from the tablet, the distance
0021  C    to the edges is computed and then translated into a corresponding
0022  C    distance in the coordinate system of the chart. This method is
0023  C    preferred over the user inputting the scale factors of the chart
0024  C    because any correction neccessary for chart shrinkage is auto-
0025  C    matically compensated for.
0026  C
0027  C       DIGBT is written in FORTRAN IV and is implemented on the HP
0028  C    9640A computer located at the Engineering Department, Montana
0029  C    State University, Bozeman, MT. The computer runs under the HP
0030  C    RTE-III operating system and uses a TEKTRONIX 4954 digitizing
0031  C    tablet and a TEKTRONIX 4014 graphics terminal which are both
0032  C    controled by a modified version of the TEKTRONIX PLOT-10 Terminal
0033  C    Control System.
0034  C
0035  C       This program refers to the following support libraries:
0036  C
0037  C          DIGLB -- Digitizer tablet control routines
0038  C          GSFIO -- GEOSCAN data file I/O routines
0039  C          F2FIO -- File I/O utility routines
0040  C          TRMIO -- Terminal I/O utility routines
0041  C          STRNG -- String library
0042  C
0043  C       The following are the global variables used by DIGBT:
0044  C
0045        BLOCK DATA
0046  $INSERT /TKTRX::25     /* included here for HP implementation only */
0047  $INSERT /DIGBT::25     /* DIGBT program variables */
0048        END
0049  C.F
0050  C-------------------------------------------------------------------------
0051  C
0052  C            Digitizing program for GEOSCAN system
```

```
0053  C
0054        SUBROUTINE DIGBT(UNITS,ZUNIT,ISTA,AREA,II,NZ)
0055  C
0056  C - Global variables:
0057  $INSERT /TKTRX::25      /* TEKTRONIX PLOT-10 variables */
0058        COMMON IIX,IIY
0059  $INSERT /DIGBT::25      /* DIGBT program variables */
0060  $INSERT /DGVAR::25      /* Digitizer variables  for DIGLB */
0061  $INSERT /F2VAR::25      /* File I/O (version 2.0) variables */
0062  C
0063  C - Global constants:
0064  $INSERT /PSTAT::25      /* Procedure status constants */
0065  $INSERT /FSTAT::25      /* File I/O (version 2.0) constants */
0066  $INSERT /DSTAT::25      /* Digitizer routine status constants */
0067  $INSERT /DEVCE::25      /* Device codes for DIGLB */
0068  $INSERT /DCODS::25      /* Digitizer/display codes */
0069  $INSERT /PROGW::25      /* DISPK graphic output coordinate plane */
0070  $INSERT /TRMIO::25      /* Terminal I/O constants & variables */
0071  $INSERT /STRNG::25      /* String status constants */
0072  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0073        REAL AREA(50)
0074        INTEGER II(50),NZ(5,5)
0075        INTEGER UNITS,ZUNIT(5),ISTA(5)
0076        INTEGER FILNM(8)
0077  C
0078  C - Local variables:
0079  C
0080  C * Begin DIGBT
0081  C
0082  C
0083        CALL INITT(45)
0084        CALL INIT
0085        CALL SETUP(FILNM)
0086        CALL QUEST(FILNM,UNITS,ZUNIT,ISTA,AREA,II,NZ)
0087        RETURN
0088        END
0089  C.F
0090  C-------------------------------------------------------------------------
0091  C
0092  C        Initiation routine for DIGBT
0093  C
0094        SUBROUTINE  INIT
0095  C
0096  C - Global variables:
0097  $INSERT /DIGBT::25      /* DIGBT program variables */
0098  C
0099  C - Global constants:
0100  $INSERT /PSTAT::25      /* Procedure status constants */
0101  $INSERT /FSTAT::25      /* File I/O (version 2.0) constants */
0102  C
0103  C - Functions:
0104        INTEGER  DINIT
0105        INTEGER  FOPEN
0106  C
```

```
0107  C - Local constants:
0108        REAL  SXMIN, SYMIN,      /* CRT screen coordinates for
0109       &      SXMAX, SYMAX       /* tablet monitor area
0110        INTEGER  NOTHNG
0111  C
0112        DATA  SXMIN /300.0/,  SYMIN /125.0/,
0113       &      SXMAX /950.0/, SYMAX /625.0/
0114  C
0115  C * Begin  INIT
0116  C     /* Initialize DIGBT common */
0117        SCRFIL = 0
0118        OUTFIL = 0
0119        GFTYPE = 0
0120  C
0121        X1PNT = 0.0
0122        Y1PNT = 0.0
0123  C
0124  C     /* Initialize tablet and CRT */
0125        IF (DINIT(SXMIN,SYMIN,SXMAX,SYMAX) .NE. OK) CALL ABEND
0126        CALL CHSIZ(4)       /* Select small character size on TEK scope
0127  C
0128  C
0129  C     /* Setup scratch file */
0130        IF (FOPEN(SCRFIL,'$$SCR$::25',1,10,OLDFIL+NEWFIL,READIT+WRITIT)
0131       &    .NE. OK) CALL ABEND
0132  C
0133        CALL RESET
0134  C
0135        RETURN
0136        END
0137  C.F
0138  C-------------------------------------------------------------------------
0139  C
0140  C        Reset variables
0141  C
0142        SUBROUTINE  RESET
0143  C
0144  C - Global variables and constants:
0145  $INSERT /DIGBT::25      /* DIGBT program variables */
0146  $INSERT /F2FIO::25      /* File I/O (version 2.0) constants */
0147  C
0148  C - Local variables:
0149        REAL  SMALL, HUGE
0150  C
0151        DATA  SMALL /-1E36/,  HUGE /1E36/
0152  C
0153  C * Begin RESET
0154        NPTS = 0
0155        NCOL = 2
0156  C
0157        FXMIN = HUGE
0158        FYMIN = HUGE
0159        FXMAX = SMALL
0160        FYMAX = SMALL
```

```
0161  C
0162        CALL FRWND (SCRFIL)
0163  C
0164        RETURN
0165        END
0166  C.F
0167  C-------------------------------------------------------------------------
0168  C
0169  C       Setup routine for DIGBT
0170  C
0171        SUBROUTINE  SETUP(FILNM)
0172  C
0173  C - Global variables:
0174  $INSERT /DIGBT::25      /* DIGBT program variables */
0175  $INSERT /DGVAR::25      /* Digitizer variables  for DIGLB */
0176  C
0177  C - Global constants:
0178  $INSERT /PSTAT::25      /* Procedure status constants */
0179  $INSERT /FSTAT::25      /* File I/O (version 2.0) constants */
0180  $INSERT /DSTAT::25      /* Digitizer routine status constants */
0181  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0182  $INSERT /TRMIO::25      /* Terminal I/O constants */
0183  $INSERT /STRNG::25      /* String status constants */
0184  C
0185  C - Functions:
0186        INTEGER  FOPEN
0187        INTEGER  DIGI
0188        INTEGER  LEN, TLEN, CMPAR
0189  C
0190  C - Local variables:
0191        INTEGER  DSTATS    /* Digitizer routine status
0192        INTEGER  ANSWER(40), ANSLEN, MAXAL     /* User answer string
0193        INTEGER  TITLE(40), TITLEN, MAXTL      /* Title string
0194        INTEGER FILNM(8)
0195        REAL  XLLVAL, YLLVAL,     /* Lower left coordinate of chart
0196       &      XURVAL, YURVAL      /* Upper right coordinate of chart
0197        LOGICAL  ABORT            /* TRUE if user wants to abort
0198  C
0199        DATA  MAXAL /80/,  MAXTL /80/
0200  C
0201  C * Begin SETUP
0202        CALL FILL (ANSWER,1,MAXAL,' ')
0203        CALL ASK('TYPE 6 CHARACTER OUTPUT FILE NAME,THEN ::25 ',44,ANSWER
0204       &,MAXAL)
0205  C
0206        DO 2,I=1,8
0207        FILNM(I) = ANSWER(I)
0208   2    CONTINUE
0209        IF (FOPEN (OUTFIL,ANSWER,1,TLEN(ANSWER,1,MAXAL,' '),NEWFIL,WRITIT)
0210       &           .NE. OK) CALL ABEND
0211  C
0212        CALL FILL (TITLE,1,MAXTL,' ')
0213        CALL ASK ('Type a title for the data file -> ',34,TITLE,MAXTL)
0214        TITLEN = TLEN (TITLE,1,MAXTL,' ')
```

```
0215  C
0216            GFTYPE = AREAF
0217  C
0218   4     CALL ASK ('What is the X coordinate of lower left corner? ',47,
0219        &           XLLVAL,REALV)
0220         CALL ASK ('What is the Y coordinate of lower left corner? ',47,
0221        &           YLLVAL,REALV)
0222         CALL ASK ('What is the X coordinate of upper right corner? ',48,
0223        &           XURVAL,REALV)
0224         CALL ASK ('What is the Y coordinate of upper right corner? ',48,
0225        &           YURVAL,REALV)
0226  C
0227         CALL SAY ('Now, from the tablet:',21)
0228  C
0229   5     DSTATS = NOTVAL
0230   51    IF (.NOT. (DSTATS .EQ. NOTVAL) ) GOTO 7
0231            CALL SAY ('Digitize the lower left, then the',33)
0232            CALL SAY ('upper right corners of the map.',31)
0233            DSTATS = DIG2I (XLLVAL,YLLVAL,XURVAL,YURVAL)
0234            IF (DSTATS .EQ. ERR) CALL ERROR (PRGERR,'SETUP--@51, DSTATS= ',
0235        &                                     19,DSTATS,INTEGR)
0236            IF (.NOT. (DSTATS .EQ. NOTVAL) ) GOTO 59
0237               CALL ASK ('Do you want to quit? ',21,ABORT,LOGICL)
0238               IF (ABORT) CALL ABEND
0239               CALL SAY('Then use the cursor/pen on the tablet for input.'
0240        &                 ,48)
0241   59       GOTO 51
0242  C
0243  C      /* Setup file header */
0244   7     CALL WFHED (OUTFIL,GFTYPE,'DIGBT',5,TITLE,TITLEN,
0245        &            XLLVAL,YLLVAL,XURVAL,YURVAL)
0246  C
0247         RETURN
0248         END
0249  C.F
0250  C-------------------------------------------------------------------------
0251  C
0252  C       Digitizing routine for DIGBT
0253  C
0254         SUBROUTINE  DIGTZ
0255  C
0256  C - Global variables and constants:
0257  $INSERT  /TKTRX::25     /* TEKTRONIX PLOT VARIABLES */
0258           COMMON IIX,IIY
0259  $INSERT /DIGBT::25      /* DIGBT program variables */
0260  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0261  $INSERT /DIGLB::25      /* Digitizer constants & variables */
0262  $INSERT /TRMIO::25      /* Terminal I/O constants */
0263  C
0264  C - Functions:
0265         INTEGER  DIG
0266  C
0267  C - Local variables:
0268         LOGICAL  DONE      /* TRUE if user finished
```

```
0269          INTEGER  DSTATS    /* Digitizer routine status
0270          REAL  XX, YY       /* Coordinate point
0271    C
0272    C * Begin DIGTZ
0273          DONE = .FALSE.
0274    C
0275          WRITE(7,100)
0276     1    IF (DONE) GOTO 9
0277             IF (NPTS .EQ. 0) FIRSTP = .TRUE.
0278             DSTATS = DIG (XX,YY)
0279             IF (.NOT. (DSTATS .EQ. OK) ) GOTO 2
0280                CALL WFDAT (SCRFIL,XX,YY)
0281                NPTS = NPTS + 1
0282                IF (.NOT. (NPTS .EQ. 1) ) GOTO 12
0283                   X1PNT = XX
0284                   Y1PNT = YY
0285    12          FXMIN = AMIN1(XX,FXMIN)
0286                FYMIN = AMIN1(YY,FYMIN)
0287                FXMAX = AMAX1(XX,FXMAX)
0288                FYMAX = AMAX1(YY,FYMAX)
0289                GOTO 7
0290    C
0291     2       IF (.NOT. (DSTATS .EQ. NOTVAL) ) GOTO 3
0292                CALL OPINP (DONE)
0293          IF (.NOT. DONE) GO TO 9
0294                GOTO 7
0295    C
0296     3       IF (.NOT. (DSTATS .EQ. OUTBND) ) GOTO 4
0297    C
0298          CALL BELL
0299          CALL BELL
0300          WRITE(7,110)
0301             GOTO 7
0302    C
0303     4    IIY=IIY-12
0304          CALL MOVAR(0,IIY)
0305          CALL ANMOD
0306               CALL ERROR (PRGERR,'DIGTZ-@4, DSTATS= ',18,DSTATS,INTEGR)
0307    C
0308     7       GOTO 1
0309    C
0310     9    RETURN
0311     100  FORMAT(' BEGIN DIGITIZING')
0312     110  FORMAT(' POINT IS OUT OF BOUNDS TRY AGAIN.')
0313          END
0314    C.F
0315    C-------------------------------------------------------------------------
0316    C
0317    C       Option input routine for DIGBT
0318    C
0319          SUBROUTINE OPINP (DONE)
0320          LOGICAL  DONE
0321    C
0322    C - Global variables:
```

```
0323  $INSERT  /TKTRX::25     /* TECTRONICS VARIABLES*/
0324        COMMON IIX,IIY
0325  $INSERT /DIGBT::25      /* DIGBT program variables */
0326  C
0327  C - Global constants:
0328  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0329  $INSERT /TRMIO::25      /* Terminal I/O constants */
0330  $INSERT /STRNG::25      /* String status constants */
0331  C
0332  C - Functions:
0333        INTEGER  LEN, CMPAR
0334  C
0335  C - Local variables:
0336        INTEGER  ANSWER(20), ANSLEN, MAXAL   /* User answer string
0337        LOGICAL  ENDED     /* TRUE if area boundry ended (closed)
0338        LOGICAL  DELETE
0339  C
0340        DATA  MAXAL /40/
0341  C
0342  C * Begin OPINP
0343        DONE = .FALSE.
0344        ENDED = .FALSE.
0345  C
0346   1    DELETE = .FALSE.
0347  C
0348        WRITE(7,100)
0349        READ(7,110) IANS
0350  C
0351        IF (IANS.EQ.1HY) DELETE = .TRUE.
0352  C
0353        IF (.NOT. DELETE) GO TO 44
0354  C
0355  C
0356           CALL DELPT
0357           GOTO 1
0358  C
0359   44   IF (.NOT. (NPTS .GT. 1)) GO TO 41
0360        CALL ADDPT(X1PNT,Y1PNT)
0361        ENDED=.TRUE.
0362   41   CALL RECAP
0363        CALL RESET
0364        RETURN
0365  C
0366   100  FORMAT(' DELETE LAST POINT (Y/N) ? _')
0367   110  FORMAT(1A1)
0368  C
0369        END
0370  C
0371  C.F
0372  C-------------------------------------------------------------------------
0373  C
0374  C      Add a point to data file for DIGBT
0375  C
0376        SUBROUTINE  ADDPT (XX,YY)
```

```
0377        REAL  XX, YY
0378  C
0379  C - Global variables & constants:
0380  $INSERT /DIGBT::25      /* DIGBT program variables */
0381  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0382  $INSERT /DIGLB::25      /* Digitizer constants & variables */
0383  C
0384  C - Local variables:
0385        REAL  XXPNT, YYPNT      /* Display coordinate point
0386  C
0387  C * Begin ADDPT
0388  C     /* Display point on CRT screen */
0389        CALL TRANS (WDATA,WOUT,XX,YY,XXPNT,YYPNT)
0390  C
0391   2        CALL DISPT (TKTERM,WOUT,DRAW,XXPNT,YYPNT)
0392  C
0393   5    CALL WFDAT (SCRFIL,XX,YY)
0394        NPTS = NPTS + 1
0395        IF(.NOT. (NPTS .EQ. 1) ) GOTO 7
0396           X1PNT = XX
0397           Y1PNT = YY
0398  C
0399   7    FXMIN = AMIN1(XX,FXMIN)
0400        FYMIN = AMIN1(YY,FYMIN)
0401        FXMAX = AMAX1(XX,FXMAX)
0402        FYMAX = AMAX1(YY,FYMAX)
0403  C
0404        RETURN
0405        END
0406  C.F
0407  C-------------------------------------------------------------------------------
0408  C
0409  C             Delete a data point for DIGBT
0410  C
0411        SUBROUTINE DELPT
0412  C
0413  C - Global variables and constants:
0414  $INSERT  /TKTRX::25    /*TECKTRONICS  VARIABLES*/
0415        COMMON IIX,IIY
0416  $INSERT /PSTAT::25      /* Procedure status constants */
0417  $INSERT /DIGBT::25      /* DIGBT program variables */
0418  $INSERT /DGVAR::25      /* Digitizer variables  for DIGLB */
0419  $INSERT /DCODS::25      /* Digitizer/display codes */
0420  $INSERT /TRMIO::25      /* Terminal I/O constants */
0421  C
0422  C - Functions:
0423        INTEGER  RFDAT
0424  C
0425  C - Local variables:
0426        REAL  XX, YY,         /* Data coordinate
0427       &      XXPNT,YYPNT    /* Display coordinate point
0428  C
0429  C * Begin DELPT
0430        IF (.NOT. (NPTS .GT. 0) ) GOTO 8
```

```
0431            IF (.NOT. (RPDAT (SCRFIL,XX,YY) .EQ. OK) ) GOTO 9
0432               CALL TRANS (WDATA,WOUT,XX,YY,XXPNT,YYPNT)
0433               CALL DISPT (TKTERM,WOUT,MOVE,XXPNT,YYPNT)
0434               CALL XDRW (TKTERM,WOUT,6,0)
0435               IF (XX .EQ. FXMIN) RECHK = .TRUE.
0436               IF (XX .EQ. FXMAX) RECHK = .TRUE.
0437               IF (YY .EQ. FYMIN) RECHK = .TRUE.
0438               IF (YY .EQ. FYMAX) RECHK = .TRUE.
0439               NPTS = NPTS - 1
0440               IF (.NOT.(NPTS .GT. 0) ) GOTO 6
0441                  IF (.NOT. (RPDAT (SCRFIL,XX,YY) .EQ. OK) ) GOTO 6
0442                     CALL TRANS (WDATA,WOUT,XX,YY,XXPNT,YYPNT)
0443                     CALL DISPT (TKTERM,WOUT,MOVE,XXPNT,YYPNT)
0444                     CALL WFDAT (SCRFIL,XX,YY)
0445                     GOTO 9
0446  C
0447   6              CALL RESET
0448                  FIRSTP = .TRUE.
0449                  GOTO 9
0450  C
0451   8     IIY=IIY-12
0452         CALL MOVAB(0,IIY)
0453         CALL ANMOD
0454           CALL ERROR (USRERR,'No points left to delete.',25,0,NULL)
0455  C
0456   9     RETURN
0457         END
0458  C.F
0459  C-------------------------------------------------------------------------
0460  C
0461  C       Transfer summarized data to output file routine for DIGBT
0462  C
0463         SUBROUTINE  RECAP
0464  C
0465  C - Global variables & constants:
0466  $INSERT /DIGBT::25      /* DIGBT program variables */
0467  $INSERT /DGVAR::25      /* Digitizer variables  for DIGLB */
0468  $INSERT /GFTYP::25      /* GEOSCAN data file types */
0469  $INSERT /F2FIO::25      /* File I/O (version 2.0) constants */
0470  C
0471  C - Functions:
0472         INTEGER  FRWND, FWRIT, RFDAT
0473  C
0474  C - Local variables:
0475         INTEGER  I
0476         REAL  XX, YY  /* Coordinate point
0477         REAL  HUGE, SMALL
0478  C
0479         DATA  HUGE /10E36/,  SMALL /-10E36/
0480  C
0481  C * Begin RECAP
0482         IF (FRWND(SCRFIL) .NE. OK) CALL ABEND
0483         IF (.NOT. (RECHK) ) GOTO 2
0484            FXMIN = HUGE
```

```
0485            FYMIN = HUGE
0486            FXMAX = SMALL
0487            FYMAX = SMALL
0488            DO 11 I=1,NPTS
0489               IF (.NOT. (RFDAT (SCRFIL,XXIN,YYIN) .EQ. OK) ) CALL ABEND
0490               FXMIN = AMIN1(XXIN,FXMIN)
0491               FYMIN = AMIN1(YYIN,FYMIN)
0492               FXMAX = AMAX1(XXIN,FXMAX)
0493               FYMAX = AMAX1(YYIN,FYMAX)
0494   11          CONTINUE
0495   C
0496            RECHK = .FALSE.
0497            IF (FRWND(SCRFIL) .NE. OK) CALL ABEND
0498   C
0499    2     IF (.NOT. (NPTS .GT. 0) ) GOTO 9
0500            DMIN(1) = FXMIN
0501            DMIN(2) = FYMIN
0502            DMAX(1) = FXMAX
0503            DMAX(2) = FYMAX
0504            CALL WSHED (OUTFIL,GFTYPE,NPTS,NCOL,DMIN,DMAX)
0505            DO 21 I=1,NPTS
0506               IF (.NOT. (RFDAT (SCRFIL,XX,YY) .EQ. OK) ) CALL ABEND
0507               CALL WFDAT (OUTFIL,XX,YY)
0508   21          CONTINUE
0509   C
0510          CALL WSEOD (OUTFIL)
0511   C
0512    9     RETURN
0513          END
0514   C.F
0515   C-------------------------------------------------------------------------------
0516   C
0517   C       Cease routine for DIGBT
0518   C
0519          SUBROUTINE  CEASE
0520   C
0521   C - Global constants and variables:
0522   $INSERT  /TKTRX::25  /*TECKTRONICS VARIABLES*/
0523          COMMON IIX,IIY
0524   $INSERT /DIGBT::25      /* DIGBT program variables */
0525   $INSERT /F2FIO::25      /* File I/O (version 2.0) constants */
0526   C
0527   C - Local variables:
0528          INTEGER  NOTHNG
0529   C
0530   C * Begin CEASE
0531          CALL WFEOF (OUTFIL)
0532          CALL FCLOS (OUTFIL,SAVE)
0533          CALL FCLOS (SCRFIL,DELETE)
0534          RETURN
0535          END
0536   END$
```

```
&SORTX T=00004 IS ON CR00025 USING 00006 BLKS R=0014

0001  FTN4
0002        PROGRAM SORTX
0003        COMMON ITEK(45)
0004        COMMON UNITS,ZUNIT(5),ISTA(5),I1,I2,I3
0005        REAL AUD(5),AZD(5,5),ASD(5,5,20)
0006        REAL AU(5),AZ(5,5),AS(5,5,20)
0007        INTEGER NZD(5,5),IID(50)
0008        INTEGER NZ(5,5),COMMON,BUFL,DDA,II(50),DA
0009        INTEGER ZUNIT,UNITS,ISTA
0010        DIMENSION COMMON(59),AREA(50),IPRM(5)
0011        REAL FELLNC(5,5),PR(5,5),FELLTM(5,5)
0012        REAL FNCDUM(5,5),PRDUM(5,5),FTMDUM(5,5)
0013        INTEGER BUF(1285),TRKSAV
0014        EQUIVALENCE(BUF(1),FNCDUM(1,1)),(BUF(51),PRDUM(1,1)),
0015       & (BUF(101),FTMDUM(1,1)),(BUF(151),AUD(1)),
0016       &(BUF(161),AZD(1,1)),(BUF(211),ASD(1,1,1)),
0017       &(BUF(1211),IID(1)),(BUF(1261),NZD(1,1))
0018        EQUIVALENCE (COMMON(1),ITEK(1)),
0019       &(COMMON(57),ISTRK),(COMMON(58),ISECT),(COMMON(59),DDA)
0020        CALL RMPAR(IPRM)
0021        BUFL = 59
0022        CALL EXEC (14,1,COMMON,BUFL)
0023        TRKSAV = ISTRK
0024        CALL SORT(FELLNC,PR,FELLTM,DA,AU,AZ,AS,II,NZ)
0025        DDA=DA
0026  C
0027        DO 30 I=1,5
0028  C
0029           AUD(I) = AU(I)
0030  C
0031           DO 40 J=1,5
0032  C
0033              FNCDUM(I,J) = FELLNC(I,J)
0034              FTMDUM(I,J) = FELLTM(I,J)
0035              PRDUM(I,J) = PR(I,J)
0036              AZD(I,J) = AZ(I,J)
0037              NZD(I,J) =    NZ(I,J)
0038  C
0039              DO 50 K=1,20
0040  C
0041                 ASD(I,J,K) = AS(I,J,K)
0042  C
0043   50         CONTINUE
0044   40      CONTINUE
0045   30   CONTINUE
0046  C
0047        DO 60 I=1,50
0048           IID(I) = II(I)
0049   60   CONTINUE
0050  C
0051        BUFL=59
0052        CALL EXEC(14,2,COMMON,BUFL)
```

```
0053          BUFL = 1285
0054          ISTRK = TRKSAV
0055          ISECT = 0
0056          CALL EXEC(2,66,BUF,BUFL,ISTRK,ISECT)
0057          END
```

```
&SORT  T=00004 IS ON CR00025 USING 00048 BLKS R=0125

0001  FTN4
0002        SUBROUTINE SORT(FELLNC,PR,FELLTM,DA,AU,AZ,AS,II,NZ)
0003  C  THIS PROGRAM RETRIEVES STAND DATA FROM DISC
0004  C  AND ADJUSTS IT TO REFLECT CUTTING UNIT AND
0005  C  ZONE BOUNDARIES.  A REVISED SET OF DATA
0006  C  (SPECIE,HEIGHT,DBH,#TREES/ACRE) IS CREATED
0007  C  FOR EACH ZONE.
0008  C  THE INPUT DATA RE ACRES CAN BE INPUT AT THE TERMINAL
0009  C  OR BY DIGITIZING THE AREA WHEN REQUESTED.
0010        COMMON ITEK(45)
0011        COMMON UNITS,ZUNIT(5),ISTA(5),I1,I2,I3
0012        INTEGER UNITS,ISTA,DA,NZ(5,5),II(50)
0013        INTEGER COMMON(1),IPRM(5),ZUNIT
0014        REAL FELLNC(5,5),PR(5,5),FELLTM(5,5),W(13)
0015        REAL AU(5),AZ(5,5),AS(5,5,20)
0016        INTEGER FOR,RD,DR,SD,DIGB(3),BUF(1),AD,DA,S(25)
0017        EQUIVALENCE (COMMON(1),ITEK(1))
0018        DIMENSION ACRE(99),H(25),D(25),T(25),NAME(3),IFOR(20),
0019       1NUMB(10),Q(5,5),SS(5,5,25),HH(5,5,25),DD(5,5,25)
0020       2,TT(5,5,25),NK(10),AREA(25)
0021  C  IFOR (J) & NUMB(I) ARE ARRAYS FOR TABLE LOOKUPS IN HEADER INFO.
0022        DATA IFOR/2HA ,2HB ,2HC ,2HD ,2HE ,2HF ,2HG ,2HH ,2HI ,
0023       &2HJ ,2HK ,2HL ,2HM ,2HN ,2HO ,2HP ,2HQ ,2HR ,2HS ,2HT /
0024        DATA NUMB/2H0 ,2H1 ,2H2 ,2H3 ,2H4 ,2H5 ,2H6 ,2H7 ,2H8 ,2H9 /
0025        DATA DIGB /2HDI,2HGB,2HT /
0026  C  ISLP=AVERAGE SLOPE
0027  C  METH=INTEGRATED OR DISJOINT
0028  C  ICOND=FAVORABLE OR UNFAVORABLE
0029  C  IOPT=BRANCH TO OPTIMIZATION ROUTINE?
0030  C  NDECK=NUMBER OF DECKS PER ZONE
0031  C  IDEK=SINGLE OR MULTIPLE DECKS
0032        NUNITS=UNITS
0033        CALL NEWPG
0034        WRITE(9,600)
0035   600  FORMAT(1X,'ENTER THE NUMBER CORRESPONDING TO THE DESIRED OPTION',
0036       &//,'*************************************************************'
0037       &,'*************',///)
0038        WRITE(9,601)
0039  601   FORMAT(1X,'1 = DIGITIZE TO DETERMINE AREAS & USE ACTUAL STAND',
0040       &' DATA FOR CALCULATIONS',///,1X,
0041       &'2 = INPUT AVERAGES FOR AREAS & FIELD DATA',
0042       &' FROM THE TERMINAL',///,
0043       &'*************************************************',
0044       &'******************',//,' ----------------------> _')
0045        READ(9,*)DA
0046        CALL NEWPG
0047        IF(DA.EQ.2) GO TO 602
0048        CALL DIGBT (AU,AZ,AS,II,NZ)
0049        GO TO 610
0050  C  INPUT AREAS DIRECTLY FROM TERMINAL
0051   602  L=0
0052        N=0
```

```
0053          DO 607 I=1,NUNITS
0054          N=N+1
0055          WRITE(9,416)I
0056     416  FORMAT(1X,'ENTER THE TOTAL NUMBER OF ACRES IN CUTTING UNIT ',I2
0057          READ(9,*)AU(I)
0058          WRITE(9,409)I
0059     409  FORMAT(1X,'DO YOU DESIRE TO PARTITION CUTTING UNIT ',I2,' INTO',
0060         1,/,' ZONES TO DISTINGUISH DIFFERENT TOPOGRAPHY? ',/,
0061         2' YE = YES, NO = NO ---> _')
0062          READ(9,410)MI
0063     410  FORMAT(A2)
0064          IF(MI.EQ.2HNO)GO TO 417
0065          WRITE(9,411)
0066     411  FORMAT(1X,'HOW MANY ZONES ARE DESIRED? ---> _')
0067          READ(9,*)ZUNIT(I)
0068          GO TO 418
0069     417  ZUNIT(I)=1
0070     418  DO 607 J=1,ZUNIT(I)
0071          N=N+1
0072          WRITE(9,605)J,I
0073     605  FORMAT(' ENTER THE TOTAL ACRES IN ZONE ',I2,' OF CUTTING UNIT ',
0074         1I2)
0075          READ(9,*)AZ(I,J)
0076          WRITE(9,606)J,I
0077     606  FORMAT(1X,'ENTER THE NUMBER OF STANDS TO BE LOGGED IN ZONE ',I2,
0078         &' OF UNIT ',I2)
0079          READ(9,*)NZ(I,J)
0080          DO 607 K=1,NZ(I,J)
0081          L=L+1
0082          N=N+1
0083          WRITE(9,608)J
0084     608  FORMAT(1X,'ENTER STAND NUMBER FOLLOWED BY STAND AREA THAT LIES'
0085         1,/,' WITHIN ZONE ',I2,' SEPARATE DATA WITH COMMA.')
0086          READ(9,*)II(L),AS(I,J,K)
0087     607  CONTINUE
0088     C    IF(DA.EQ.2) RETURN
0089     C  INPUT STAND IDENTIFICATION & LOCATE STAND DATA FOR EACH AREA
0090      610  JN=0
0091          DO 655 MMM=1,NUNITS
0092          DO 655 NNN=1,ZUNIT(MMM)
0093          FELLNC(MMM,NNN)=0.
0094     655  CONTINUE
0095          DO 412 I=1,NUNITS
0096          WRITE(9,51)I
0097     51   FORMAT(//,1X,'THE FOLLOWING CONCERNS CUTTING UNIT NO ',I2,//)
0098     62   WRITE(9,1)
0099     1    FORMAT(1X,'ENTER THE NUMBER OF DIGITS INDICATED',//,
0100         11X,'FOREST NUMBER = _')
0101          READ(9,*)FOR
0102          WRITE(9,2)
0103     2    FORMAT(1X,'BLOCK OR RANGER DISTRICT NUMBER = _')
0104          READ(9,*)RD
0105          WRITE(9,3)
0106     3    FORMAT(1X,'COMPARTMENT OR MAIN DRAINAGE NUMBER = _')
```

```
0107          READ(9,*)DR
0108          WRITE(9,4)
0109     4    FORMAT(1X,'SUBCOMPARTMENT OR SUBDRAINAGE NUMBER = _')
0110          READ(9,*)SD
0111          DO 500 L=1,ZUNIT(I)
0112          TOT=0.
0113          K=0
0114          DO 406 J=1,NZ(I,L)
0115          JN=JN+1
0116     C
0117     C    HEADER INFORMATION:::FILE NAME WHICH IDENTIFIES FOREST,
0118     C    RANGER DISTRICT,COMPARTMENT, AND SUB-COMPARTMENT
0119     C    EXAMPLE::$K6061=GALLATIN,RD=6,COMPARTMENT=06,
0120     C    SUB-COMPARTMENT=1.    THIS IS FOLLOWED BY STAND NO.,
0121     C    ACRES, AND THE NUMBER OF ENTRIES IN THE FILE.  THEN SPECIE,
0122     C    HEIGHT,DBH, AND TREES PER ACRE FOLLOW.  THE LAST ENTRY
0123     C    IS EOF MARKER=1.E36.
0124     C
0125          ITEMP=IFOR(FOR)
0126          ITEMP=ITEMP/256
0127          IPD=2H$ /256
0128          IPD=IPD*256
0129          NAME(1)=IPD+ITEMP
0130          ITEMP=NUMB(RD+1)
0131          ITEMP=ITEMP/256
0132          ITEMP=ITEMP*256
0133          IDRT=(DR/10)+1
0134          ITMP=NUMB(IDRT)
0135          ITMP=ITMP/256
0136          NAME(2)=ITEMP+ITMP
0137          IX=(DR-((IDRT-1)*10))+1
0138          ITEMP=NUMB(IX)
0139          ITEMP=ITEMP/256
0140          ITEMP=ITEMP*256
0141          ITMP=NUMB(SD+1)
0142          ITMP=ITMP/256
0143          NAME(3)=ITEMP+ITMP
0144          CALL SPOLU(INP,NAME,1,1,25)
0145          IF (INP .LT. 0) GO TO 900
0146    407   READ(INP,65)(NAME(JM),JM=1,3),IST,ACRE(IST),NO
0147          WRITE(9,434)(NAME(JM),JM=1,3),IST,ACRE(IST),NO
0148    434   FORMAT(1X,3A2,4X,I2,4X,F5.2,4X,I3)
0149    65    FORMAT(3A2,I2,F5.2,I3)
0150          DO 71 IB=1,NO
0151          READ(INP,66)S(IB),H(IB),D(IB),T(IB)
0152    66    FORMAT(1X,3F4.0,F5.2)
0153    71    CONTINUE
0154          READ(INP,70)PPP
0155          IF(PPP.NE.1.E36)GO TO 430
0156    70    FORMAT(E6.1)
0157          IF(II(JN).NE.IST)GO TO 407
0158          CALL SPOLU(INP,NAME,1,2,25)
0159     C    FOR EACH STAND CALCULATE THE INFORMATION REQUIRED AND STORE
0160     C    IT TO BE COMBINED WITH THE DATA FROM THE NEXT STAND
```

```
0161  C
0162  C  CALCULATE THE NUMBER OF TREES PER STAND AND ACCUMULATE
0163  C  THEM ON A PER ZONE BASIS
0164        DO 503 NN=1,NO
0165        FELLNC(I,L)=T(NN)*AS(I,L,IST) +FELLNC(I,L)
0166  503   CONTINUE
0167  C  CALCULATE CYCLE TIME PER TREE= TIME TO FELL,LIMB,
0168  C  TOP,&WALK TO NEXT TREE.  ASSUME 100 TREES/DAY AND A 6 HOUR DAY
0169  C  WHICH YEILDS 3.57 MINUTES PER TREE.
0170        FELLTM(I,L)=3.57
0171  C  CALCULATE THE AVERAGE TREE SIZE IN POUNDS IN EACH ZONE.
0172  C  BUTT DIAMETER IS NEEDED FOR THE EQUATION   DBH IS GIVEN
0173  C  IN STAND TABLES.  THUS, DBH + 5/8=BUTT DIA IF THE TAPER
0174  C  IS 1" PER 8'.
0175  C  CONVERT DBH TO BUTT DIA AND CALCULATE VOLUME IN CUBIC FT
0176  C  FOR ALL TREES OF THIS DBH,HEIGHT,AND SPECIES IN THE STAND.
0177        DO 505 MZ=1,13
0178   505  W(MZ)=0.
0179        DO 30 KI=1,NO
0180  C  CONVERT HEIGHT TO MEDIAN OF INTERVAL
0181        H(KI)=H(KI)+4.95
0182        VOL=((3.1416/576.)*((D(KI)+.625)-H(KI)/16.)**2)
0183       &*H(KI)*ABS(T(KI)*AS(I,L,IST))
0184  C  IF TREE IS DEAD,REDUCE WEIGHT BY A FIXED %.
0185        IF(T(KI).LT.0.)VOL=VOL*.8
0186        GO TO (11,12,13,14,15,16,17,18,19,20,21,22,23),S(KI)
0187  C  W()=WEIGHT(LBS) FOR EACH SPECIE IN THE STAND
0188    11  W(1)=W(1)+VOL*36.
0189        GO TO 30
0190    12  W(2)=W(2)+VOL*32.
0191        GO TO 30
0192    13  W(3)=W(3)+VOL*49.
0193        GO TO 30
0194    14  W(4)=W(4)+VOL*34.
0195        GO TO 30
0196    15  W(5)=W(5)+VOL*37.
0197        GO TO 30
0198    16  W(6)=W(6)+VOL*39.
0199        GO TO 30
0200    17  W(7)=W(7)+VOL*36.
0201        GO TO 30
0202    18  W(8)=W(8)+VOL*45.
0203        GO TO 30
0204    19  W(9)=W(9)+VOL*35.
0205        GO TO 30
0206    20  W(10)=W(10)+VOL*43.
0207        GO TO 30
0208    21  W(11)=W(11)+VOL*43.
0209        GO TO 30
0210    22  W(12)=W(12)+VOL*40.
0211        GO TO 30
0212    23  W(13)=W(13)+VOL*40.
0213    30  CONTINUE
0214  C  CALCULATE THE TOTAL & AVERAGE WEIGHT(LBS) OF STAND
```

```
0215  C  TIMBER
0216        DO 31 IM=1,13
0217        PR(I,L)=PR(I,L)+W(IM)
0218   31   CONTINUE
0219        NNO=NNO+NO
0220  C  STORE EACH ADJUSTED FILE DATA INTO ARRAYS FOR USE
0221        DO 408 M=1,NO
0222        IF(S(M).EQ.0)GO TO 408
0223        K=K+1
0224  C  NK(L)=NUMBER OF ENTRIES IN DATA FILE FOR ZONE L
0225        NK(L)=K
0226        SS(I,L,K)=S(M)
0227        HH(I,L,K)=H(M)
0228        DD(I,L,K)=D(M)
0229        TT(I,L,K)=T(M)
0230   408  CONTINUE
0231   406  CONTINUE
0232   500  CONTINUE
0233        CALL NEWPG
0234        DO 424 L=1,ZUNIT(I)
0235        WRITE(9,435)(NAME(KI),KI=1,3),I,L
0236   435  FORMAT(1X,3A2,4X,'UNIT ',I2,4X,'ZONE ',I2,4X)
0237        DO 422 J=1,NK(L)
0238        WRITE(9,421)SS(I,L,J),HH(I,L,J),DD(I,L,J),TT(I,L,J)
0239   421  FORMAT(1X,4F10.2)
0240   422  CONTINUE
0241        WRITE(9,423)NK(L),L
0242   423  FORMAT(1X,'THERE ARE ',I2,' ENTRIES IN THE DATA FILE'
0243       1' FOR ZONE ',I2)
0244   424  CONTINUE
0245        WRITE(9,433)
0246   433  FORMAT(1X,//////,'PRESS RETURN KEY  TO CONTINUE')
0247        CALL TINPT(ICHAR)
0248        CALL NEWPG
0249   412  CONTINUE
0250  C  CALCULATE STATISTICS OF AREAS FOR INPUTS TO FELL,BNCH,ETC.
0251        IF(PPP.EQ.1.E36)GO TO 436
0252   430  WRITE(9,432)(NAME(IL),IL=1,3),IST,ACRE(IST),NO
0253   432  FORMAT(1X,'ERROR IN READING CONSTANT FOR EOF(1.E36) ',3A2,3X,I2,3X
0254       1,F5.2,3X,I2)
0255   436  RETURN
0256   900   WRITE (9,9000) INP
0257   9000 FORMAT(X' ERROR ',I5,' ON FILE OPEN')
0258        STOP
0259        END
```

```
&QUEST T=00004 IS ON CR00025 USING 00019 BLKS R=0008

0001  FTN
0002  C-------------------------------------------------------------------------
0003  C
0004  C          THIS PROGRAM ASKS QUESTIONS RE DIGITIZING ROUTINE FOR DIGBT
0005  C
0006  C-------------------------------------------------------------------------
0007  C
0008  C
0009        SUBROUTINE QUEST(FILNM,UNITS,ZUNIT,ISTA,AREA,II,NZ)
0010  C
0011  $INSERT     /TKTRX::25
0012        COMMON IIX,IIY
0013  $INSERT /DIGBT::25
0014  $INSERT /DGVAR::25
0015  $INSERT /F2VAR::25
0016  $INSERT /PSTAT::25
0017  $INSERT /FSTAT::25
0018  $INSERT /DSTAT::25
0019  $INSERT /DEVCE::25
0020  $INSERT /DCODS::25
0021  $INSERT /PROGW::25
0022  $INSERT /TRMIO::25
0023  $INSERT /STRNG::25
0024  $INSERT /GFTYP::25
0025        INTEGER COMMON(1),ZUNIT(5),NZ,II,ISTA(5)
0026        INTEGER UNITS
0027        INTEGER FOR,FD,DR,SU,DIGB(3),FILNM(8)
0028        EQUIVALENCE (COMMON(1),KBAUDR)
0029  C
0030        DIMENSION NZ(5,5),II(50),AREA(50)
0031  C
0032  C  NOW THE DIGITIZING
0033  C
0034  C  THE FOLLOWING IS THE INTERACTIVE DIGITIZING ROUTINE
0035        CALL BMLOC(ISTAT,IX,IY)
0036        WRITE(7,111)
0037   111  FORMAT(1X,'WE ARE NOW READY TO DIGITIZE ONE AREA AT A TIME',
0038       &/,'WHEN COMPLETED WITH EACH AREA PRESS THE SPACE THEN',
0039       &/,' THE RETURN KEYS ON THE TEKTRONIX TERMINAL')
0040
0041        NUNITS=UNITS
0042  C
0043        L=0
0044        DO 100 I=1,NUNITS
0045        WRITE(7,101)I
0046   101  FORMAT(//,1X,'DIGITIZE BONDARIES OF CUTTING UNIT ',I2)
0047        CALL DIGTZ
0048        WRITE(7,104)I
0049   104  FORMAT(1X,'DO YOU WISH TO PARTITION THE CUTTING UNIT ',I2,/,
0050       1' INTO ZONES TO DISTINGUISH DIFFERENT TOPOGRAPHY? YE=YES,NO=NO _')
0051        READ(7,105)MI
0052   105  FORMAT(A2)
```

```
0053          ZUNIT(I) = 1
0054          IF(MI.EQ.2HNO)GO TO 108
0055          WRITE(7,106)
0056    106   FORMAT(1X,'HOW MANY ZONES ARE DESIRED? _')
0057          READ(7,*)ZUNIT(I)
0058    108   DO 100 J=1,ZUNIT(I)
0059          IF (MI .EQ. 2HNO) GO TO 1090
0060          WRITE(7,109)J,I
0061    109   FORMAT(1X,'DIGITIZE BOUNDARY OF ZONE ',I2,' OF CUTTING UNIT ',I2)
0062          CALL DIGTZ
0063    1090  WRITE(7,110)J
0064    110   FORMAT(1X,'ENTER THE NUMBER OF STANDS TO BE LOGGED IN ZONE ',I2,
0065         1' --> _')
0066          READ(7,*)NZ(I,J)
0067    C
0068          DO 100 K=1,NZ(I,J)
0069          L=L+1
0070          WRITE(7,112)J,I
0071    112   FORMAT(1X,'ENTER THE STAND NUMBER OF THE NEXT STAND TO BE',/,
0072         1' DIGITIZED IN ZONE ',I2,' OF CUTTING UNIT ',I2,' --> _')
0073          READ(7,*)II(L)
0074          WRITE(7,113)II(L),J,I
0075    113   FORMAT(1X,'DIGITIZE THE PORTION OF STAND ',I2,' THAT IS ',/,
0076         1' WITHIN ZONE ',I2,' OF CUTTING UNIT ',I2)
0077          IF((J.EQ.ZUNIT(I)).AND.(II(L).EQ.NZ(I,J)))FL=3
0078          CALL DIGTZ
0079    100   CONTINUE
0080          CALL CEASE
0081          ITOT=0.
0082          DO 22 I=1,NUNITS
0083          DO 21 J=1,ZUNIT(I)
0084          ITOT=ITOT+NZ(I,J)
0085    21    CONTINUE
0086          IF (ZUNIT(I) .EQ. 1) GOTO 22
0087          ITOT=ITOT+ZUNIT(I)
0088    22    CONTINUE
0089          ITOT=ITOT+NUNITS
0090          CALL A2DBT(AREA,ITOT,FILNM)
0091    C    OUTPUT THE AREA OF THE DIGITIZED UNIT,ZONES,& STANDS
0092    C
0093          N=0
0094          CALL MOVAB(0,650)
0095          CALL ANMOD
0096    C
0097          WRITE(9,127)
0098    127   FORMAT(1X,5X,'THE CALCULATED AREAS:',///)
0099    C
0100          DO 122 I=1,NUNITS
0101          N=N+1
0102          AREA(N)=AREA(N)/43560.
0103          WRITE(9,103)I,AREA(N)
0104    103   FORMAT(///,1X,'AREA OF CUTTING UNIT ',I2,' = ',F12.4)
0105          DO 122 J=1,ZUNIT(I)
0106          IF (ZUNIT(I).EQ.1) GOTO 119
```

```
0107          N=N+1
0108          AREA(N)=AREA(N)/43560.
0109    119   WRITE(9,120)J,AREA(N)
0110    120   FORMAT(    ,1X,'AREA OF ZONE          ',I2,' - ',F12.4)
0111          DO 122 K=1,NZ(I,J)
0112          N=N+1
0113          KN=KN+1
0114          AREA(N)=AREA(N)/43560.
0115          WRITE(9,121)II(KN),AREA(N)
0116    121   FORMAT(    ,1X,'AREA OF STAND         ',I2,' - ',F12.4)
0117    122   CONTINUE
0118          WRITE(9,126)
0119    126   FORMAT(1X,///,'PRESS THE RETURN KEY TO CONTINUE')
0120          CALL TINPT(ICHAR)
0121          CALL NEWPG
0122          RETURN
0123          END
```

```
&FELL4 T=00004 IS ON CR00025 USING 00037 BLKS R=0040

0001  FTN4
0002        PROGRAM FELL
0003  C
0004  C COMPUTES THE TOTAL COST, TIME, & PRODUCTION OF ACTIVITY 1
0005        COMMON ITEK(45)
0006        INTEGER ZUNIT,UNITS,DA
0007        COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
0008       1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
0009       2PHPDY(5,5),FELLTM(5,5),PR(5,5),DA
0010        DIMENSION FELLCT(5,5),
0011       2TM(5,5),CT(5,5),CPT(5,5),TPH(5,5),CPH(5,5),
0012       3TMU(5,5),PRU(5,5),CTU(5,5),CPTU(5),TPHU(5),CPHU(5)
0013        DIMENSION TTMU(5),PPRU(5),CCTU(5),BN(5),TDAUF(5),CSCDF(5),
0014       1TFVUF(5),TMDF(5,5),SCDF(5,5),TCFVF(5,5),SHPDF(5,5),PHPDF(5,5),
0015       2SCPHF(5,5),HPWDF(5,5)
0016        INTEGER COMMON(1),IPRM(5),FELL1(3)
0017        EQUIVALENCE (COMMON(1),ITEK(1))
0018        DATA FELL1 /2HFE, 2HLL, 2H# /
0019  C
0020        CALL RMPAR (IPRM)
0021  C
0022        IBUFL=491
0023        CALL EXEC (14,1,COMMON,IBUFL)
0024  C
0025        TOTCT(1)=0.
0026        TOTPR(1)=0.
0027        TOTTM(1)=0.
0028        NUNITS=UNITS
0029  C
0030        DO 27 I=1,UNITS
0031        N= ZUNIT(I)
0032        TTMU(I)=0.
0033        PPRU(I)=0.
0034        CCTU(I)=0.
0035        CPTU(I)=0.
0036        TPHU(I)=0.
0037        CPHU(I)=0.
0038        TDAUF(I)=0.
0039        CSCDF(I)=0.
0040        TFVUF(I)=0.
0041        DO 27 J=1,N
0042        FELLCT(I,J)=0.
0043  C     FELLPR(I,J)=0.
0044        SHPDF(I,J)=0.
0045        PHPDF(I,J)=0.
0046        SCPHF(I,J)=0.
0047        TMDF(I,J)=0.
0048        SCDF(I,J)=0.
0049        TCFVF(I,J)=0.
0050        HPWDF(I,J)=0.
0051        TM(I,J)=0.
0052        CT(I,J)=0.
```

```
0053         CPT(I,J)=0.
0054         TPH(I,J)=0.
0055         CPH(I,J)=0.
0056         TMU(I,J)=0.
0057         PRU(I,J)=0.
0058         CTU(I,J)=0.
0059   C
0060    27   CONTINUE
0061         WRITE(9,50)
0062    50   FORMAT(1X,'INPUT FELLING DATA')
0063         WRITE(9,200)
0064    200  FORMAT(///,1X,'DO YOU WISH TO INPUT THE FELLING CYCLE'
0065        &,' TIMES OR USE THE DEFAULT VALUES FROM EMPIRICAL DATA ?',/,
0066        &1X,'1 = INPUT MY VALUES        2 = USE DEFAULT VALUES',/,
0067        &1X,' ---------------> _')
0068         READ(9,*)IFLG
0069         DO 16 I=1,UNITS
0070         N=ZUNIT(I)
0071         DO 16 J=1,N
0072         IF(DA.EQ.1)GO TO 102
0073         IF(IFLG.EQ.2)GO TO 102
0074         WRITE(9,10)J,I
0075    10   FORMAT(///,' FELLING CYCLE TIME(MIN)FOR ZONE ',I2,' OF UNIT ',I2,
0076        1' = _')
0077         READ(9,*)FELLTM(I,J)
0078         WRITE(9,20)J,I
0079    20   FORMAT(1X,'NUMBER OF CYCLES FOR ZONE ',I2,' OF UNIT ',I2,' = _')
0080         READ(9,*)FELLNC(I,J)
0081    102  WRITE(9,70)J,I
0082    70   FORMAT(1X,'THE NUMBER OF HOURS PER WORK DAY',
0083        1' FOR ZONE ',I2,' OF UNIT ',I2,' = _')
0084         READ(9,*) HPWDF(I,J)
0085         WRITE(9,30)J,I
0086     30  FORMAT(1X,'SYSTEM FELLING COST/HR FOR ZONE ',I2,' OF UNIT ',I2,
0087        1' = _')
0088         READ(9,*)FELLCT(I,J)
0089         WRITE(9,71)J,I
0090    71   FORMAT(1X,'THE NUMBER OF SYSTEM SET-UP HOURS PER DAY',
0091        1' FOR ZONE ',I2,' OF UNIT ',I2,' = _')
0092         READ(9,*) SHPDF(I,J)
0093         WRITE(9,72)J,I
0094    72   FORMAT(1X,'SYSTEM SET-UP COST ($/HR) FOR ZONE ',I2,
0095        1' OF UNIT ',I2,' = _')
0096         READ(9,*) SCPHF(I,J)
0097         IF(DA.EQ.1)GO TO 103
0098         WRITE(9,40)J,I
0099    40   FORMAT(1X,'AVERAGE TREE SIZE(POUNDS) FOR ZONE',
0100        1I2,' OF UNIT ',I2,' = _')
0101   C  FELLPR(I,J)=AVERAGE TREE SIZE IN POUNDS
0102   C  FELLNC(I,J)=NUMBER OF TREES(CYCLES)
0103        READ(9,*)FELLPR(I,J)
0104    103  PHPDF(I,J)=HPWDF(I,J)-SHPDF(I,J)
0105    16   CONTINUE
0106   C  CALCULATE ZONE DATA FOR EACH UNIT
```

```
0107  C  I=UNIT              J=ZONE IN UNIT I
0108  C  TM(I,J)=TIME TO FELL  ZONE J ON UNIT I
0109  C  PR(I,J)=PRODUCTION(TONS) FOR ZONE J  UNIT I
0110  C  CT(I,J)=COST TO FELL  ZONE J   UNIT I
0111  C  PHPDF(I,J)=PRODUCTIVE HOURS PER DAY
0112  C  SHPDF(I,J)=SET UP HOURS PER DAY
0113  C  SCPH(I,J)=SET UP COST PER HOUR
0114  C  HPWDF(I,J)=HOURS PER WORK DAY
0115        DO 66 I=1,UNITS
0116        N=ZUNIT(I)
0117        DO 14 J=1,N
0118        TM(I,J)=FELLNC(I,J)*FELLTM(I,J)/60.
0119        IF(DA.EQ.1)GO TO 104
0120        PR(I,J)=FELLNC(I,J)*FELLPR(I,J)/2000.
0121        GO TO 105
0122   104  FELLPR(I,J)=PR(I,J)/FELLNC(I,J)
0123        PR(I,J)=PR(I,J)/2000.
0124   105  CT(I,J)=TM(I,J)*FELLCT(I,J)
0125        WRITE(9,100)J,PR(I,J),J,FELLPR(I,J)
0126   100  FORMAT(' TOTAL WEIGHT(TONS) OF TREES IN ZONE ',I2,' = ',F12.1,//,
0127       &' AVERAGE TREE SIZE(LBS) IN ZONE ',I2,' = ',F12.1)
0128        WRITE(9,31)
0129        CALL TINPT(ICHAR)
0130  C  CALCULATE PER DAY DATA
0131  C  TMDF(I,J)=TIME TO FELL ZONE J, UNIT I IN DAYS
0132  C  SCDF(I,J)=SET UP COST FOR ZONE J, UNIT I IN $
0133  C  TCFVF(I,J)=TOTAL COST, FIXED AND VARIABLE FOR ZONE J, UNIT I
0134        TMDF(I,J)=TM(I,J)/PHPDF(I,J)
0135        IC=TMDF(I,J)+1.0
0136        XIC=IC
0137        SCDF(I,J)=SCPHF(I,J)*XIC*SHPDF(I,J)
0138        TCFVF(I,J)=CT(I,J)+SCDF(I,J)
0139  C  CALCULATE $/TON   TONS/HR $/HR  ALL BY ZONE
0140  C  CPT(I,J)=COST PER TON FOR ZONE J UNIT I
0141  C  TPH(I,J)=TONS PER HOUR FOR ZONE J UNIT I
0142  C  CPH(I,J)=COST PER HOUR FOR ZONE J UNIT I
0143        CPT(I,J)=CT(I,J)/PR(I,J)
0144        TPH(I,J)=PR(I,J)/TM(I,J)
0145        CPH(I,J)=CT(I,J)/TM(I,J)
0146  C
0147  C  CALCULATE UNIT TOTALS
0148  C  TTMU(I)=TIME FOR UNIT I
0149  C  PPRU(I)=PRODUCTION FOR UNIT I
0150  C  CCTU(I)=COST FOR UNIT I
0151        TTMU(I)=TM(I,J)+TTMU(I)
0152        PPRU(I)=PR(I,J)+PPRU(I)
0153        CCTU(I)=CT(I,J)+CCTU(I)
0154  C  CALCULATE UNIT TOTALS ON PER DAY BASIS
0155  C  TDAUF(I)=TOTAL PRODUCTIVE TIME TO FELL UNIT I
0156  C  CCTU(I)=TOTAL COST OF PRODUCTIVE TIME IN $
0157  C  CSCDF(I)=TOTAL SETUP COST FOR UNIT I
0158  C  TFVUF(I)=TOTAL FIXED AND VARIABLE COST FOR UNIT I
0159  C  PPRU(I)=PRODUCTIVITY OF UNIT I IN TONS
0160        TDAUF(I)=TMDF(I,J)+TDAUF(I)
```

```
0161          CSCDF(I)=SCDF(I,J)+CSCDF(I)
0162          TFVUF(I)=TCFVF(I,J)+TFVUF(I)
0163       14    CONTINUE
0164     C
0165     C  CALCULATE SITE TOTALS
0166     C  TOTTM(1)=TOTAL TIME FOR ACTIVITY 1
0167     C  TOTPR(1)=TOTAL PRODUCTION(TONS)FOR ACTIVITY 1
0168     C  TOTCT(1)=TOTAL COST FOR ACTIVITY 1
0169          TOTTM(1)=TTMU(I)+TOTTM(1)
0170          TOTPR(1)=TOTPR(1)+PPRU(I)
0171          TOTCT(1)=TOTCT(1)+CCTU(I)
0172     C  CALCULATE $/HOUR TONS/HOUR $/TON ALL ALL BY UNIT
0173     C  CPTU(I)=COST PER TON FOR UNIT I
0174     C  TPHU(I)=TONS PER HOUR FOR UNIT I
0175     C  CPHU(I)=COST PER HOUR FOR UNIT I
0176          CPTU(I)=CCTU(I)/PPRU(I)
0177          TPHU(I)=PPRU(I)/TTMU(I)
0178          CPHU(I)=CCTU(I)/TTMU(I)
0179       66    CONTINUE
0180     C  OUTPUT FELLING DATA BY ZONE & UNIT
0181       74    CALL NEWPG
0182          WRITE(9,73)
0183       73    FORMAT(49X,'ENTER NUMBER FOR OUTPUT DESIRED',
0184         1///,39X,'0=NONE',/,39X,'1=PRODUCTIVITY AND COSTS BASED ON ',
0185         2'PER HOUR DATA',/,39X,'2=PRODUCTIVITY AND COSTS BASED ON ',
0186         3'PER DAY BASIS',/,39X,' ------------> _')
0187          READ (9,*) IK
0188          CALL NEWPG
0189          IF (IK) 74,78,75
0190       75    GO TO (76,77),IK
0191       77    NUNITS=UNITS
0192          DO 85 I=1,NUNITS
0193          WRITE(9,24)
0194          WRITE(9,17)I
0195          WRITE(9,80)
0196       80    FORMAT(36X,'TIME',34X,'COST (DOLLARS)',14X,'PRODUCTION (TONS)')
0197          WRITE(9,86)
0198       86    FORMAT(18X,38('*'),9X,30('*'),7X,17('*'))
0199          WRITE(9,81)
0200       81    FORMAT(18X,'PRODUCTIVE',9X,'SETUP',9X,'TOTAL',
0201         19X,'PRODUCTIVE',5X,'SETUP',5X,'TOTAL')
0202          WRITE(9,88)
0203       88    FORMAT(19X,'(HR/DAY)',7X,'(HR/DAY)',7X,'(DAYS)',//)
0204          N=ZUNIT(I)
0205          DO 83 J=1,N
0206          WRITE(9,82)J,PHPDF(I,J),SHPDF(I,J),TMDF(I,J),CT(I,J),SCDF(I,J),
0207         1TCFVF(I,J),PR(I,J)
0208       82    FORMAT(2X,'ZONE(',I2,')',7X,F8.2,9X,F8.2,5X,F8.2,12X,F8.2,3X,
0209         1F8.2,2X,F8.2,11X,F8.2)
0210       83    CONTINUE
0211          WRITE(9,84)TDAUF(I),CCTU(I),CSCDF(I),TFVUF(I),PPRU(I)
0212       84    FORMAT(2X,'TOTAL',40X,F8.2,12X,F8.2,3X,F8.2,2X,F8.2,11X,F8.2,///)
0213       85    CONTINUE
0214          WRITE(9,31)
```

```
0215          CALL TINPT(ICHAR)
0216          CALL NEWPG
0217          GO TO 74
0218    76    NUNITS=UNITS
0219          DO 22 I=1,NUNITS
0220          WRITE(9,24)
0221    24    FORMAT(61X,'FELLING')
0222          WRITE(9,90)
0223    90    FORMAT(56X,'(PRODUCTIVE HOURS)')
0224          WRITE(9,17)I
0225          WRITE(9,18)
0226    18    FORMAT(32X,'HOURS',10X,'TONS',10X,'COST',10X,'$/TON',
0227         110X,'TONS/HR',10X,'$/HR',//)
0228          N=ZUNIT(I)
0229          DO 22 J=1,N
0230          WRITE(9,21)J,TM(I,J),PR(I,J),CT(I,J),CPT(I,J),TPH(I,J),CPH(I,J)
0231    21    FORMAT(14X,'ZONE(',I2,')',7X,F8.2,6X,F8.2,6X,F8.2,7X,
0232         2F8.2,8X,F8.2,7X,F8.2)
0233          IF(J.NE.N) GO TO 22
0234          WRITE(9,23)TTMU(I),PPRU(I),CCTU(I),CPTU(I),TPHU(I),CPHU(I)
0235    23    FORMAT(16X,'TOTALS',7X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,8X,
0236         3F8.2,7X,F8.2,////)
0237    22    CONTINUE
0238          WRITE(9,31)
0239    31    FORMAT(1X,'PRESS RETURN KEY TO CONTINUE')
0240          CALL TINPT(ICHAR)
0241          GO TO 74
0242    C  WRITE OUT UNIT INFO TO DATA FILE (FELL#)
0243     17   FORMAT(57X,'LOGGING UNIT(',I2,')',//)
0244     78   CALL SPOLU (LU(1),FELL1,1,1,25)
0245          DO 62 I=1,NUNITS
0246          WRITE(LU(1),61)TTMU(I),PPRU(I),CCTU(I)
0247     61   FORMAT(1X,10(F10.2,1X))
0248     62   CONTINUE
0249          DO 63 I=1,NUNITS
0250          N=ZUNIT(I)
0251          DO 65 J=1,N
0252          WRITE(LU(1),64)FELLTM(I,J),FELLNC(I,J),FELLCT(I,J),FELLPR(I,J),
0253         1HPWDF(I,J),PHPDF(I,J),SHPDF(I,J),SCPHF(I,J),TMDF(I,J)
0254     64   FORMAT(9(F8.2,1X))
0255     65   CONTINUE
0256     63   CONTINUE
0257    C
0258          CALL SPOLU(LU(1),FELL1,1,2,25)
0259          CALL EXEC(14,2,COMMON,IBUFL)
0260          END
```

```
&BNCH4 T=00004 IS ON CR00025 USING 00033 BLKS R=0000

0001  FTN4
0002        PROGRAM BNCH
0003  C
0004  C COMPUTES THE TOTAL COST, TIME, & PRODUCTION OF ACTIVITY 2
0005        COMMON ITEK(45)
0006        INTEGER ZUNIT,UNITS
0007        COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
0008       1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
0009       2PHPDY(5,5)
0010        DIMENSION BUNTM(5,5),BUNCT(5,5),BCNT(5,5),
0011       2TM(5,5),PR(5,5),CT(5,5),CPT(5,5),TPH(5,5),CPH(5,5),
0012       3TMU(5,5),PRU(5,5),CTU(5,5),CPTU(5),TPHU(5),CPHU(5)
0013        DIMENSION TTMU(5),PPRU(5),CCTU(5),BN(5),HPWDB(5,5),SHPDB(5,5),
0014       1SCPHB(5,5),PHPDB(5,5),TMDB(5,5),SCDB(5,5),TCFVB(5,5),TDAUB(5),
0015       2CSCDB(5),TFVUB(5)
0016        INTEGER COMMON(1),IPRM(5),BNCH1(3)
0017        EQUIVALENCE (COMMON(1),ITEK(1))
0018  C
0019        DATA BNCH1/ 2HBN,2HCH,2H# /
0020        CALL RMPAR (IPRM)
0021  C
0022        IBUFL=390
0023        CALL EXEC (14,1,COMMON,IBUFL)
0024  D     WRITE(6,9)(COMMON(I),I=1,390)
0025  D9    FORMAT('INSIDE BNCH COMMON IS',//,
0026  D    &49(10(2X,I8),/))
0027  C
0028        TOTCT(2)=0.
0029        TOTPR(2)=0.
0030        TOTTM(2)=0.
0031        NUNITS=UNITS
0032  C
0033        DO 27 I=1,UNITS
0034        N= ZUNIT(I)
0035        TTMU(I)=0.
0036        PPRU(I)=0.
0037        CCTU(I)=0.
0038        CPTU(I)=0.
0039        TPHU(I)=0.
0040        CPHU(I)=0.
0041        TDAUB(I)=0.
0042        CSCDB(I)=0.
0043        TFVUB(I)=0.
0044        BN(I)=0.
0045        DO 27 J=1,N
0046        HPWDB(I,J)=0.
0047        PHPDB(I,J)=0.
0048        SHPDB(I,J)=0.
0049        SCPHB(I,J)=0.
0050        TMDB(I,J)=0.
0051        SCDB(I,J)=0.
0052        TCFVB(I,J)=0.
```

```
0053          BUNTM(I,J)=0.
0054          BUNCT(I,J)=0.
0055          TM(I,J)=0.
0056          BNZN(I,J)=0.
0057          PR(I,J)=0.
0058          CT(I,J)=0.
0059          CPT(I,J)=0.
0060          TPH(I,J)=0.
0061          CPH(I,J)=0.
0062          TMU(I,J)=0.
0063          PRU(I,J)=0.
0064          CTU(I,J)=0.
0065    27    CONTINUE
0066          WRITE(9,50)
0067    50    FORMAT(1X,'INPUT THE BUNCHING DATA')
0068          DO 16 I=1,UNITS
0069          N=ZUNIT(I)
0070          DO 16 J=1,N
0071          WRITE(9,10)J,I
0072    10    FORMAT(///,1X,'BUNCH CYCLE TIME(MIN)FOR ZONE',I2,' OF UNIT',I2,
0073         1'=_')
0074          READ(9,*)BUNTM(I,J)
0075          WRITE(9,70)J,I
0076    70    FORMAT(1X,'THE NUMBER OF HOURS PER WORK DAY FOR ZONE',
0077         1I2,' OF UNIT',I2,'=_')
0078          READ(9,*)HPWDB(I,J)
0079          WRITE(9,30)J,I
0080    30    FORMAT(1X,'SYSTEM BUNCHING COST/HR FOR ZONE',I2,' OF UNIT',I2,
0081         1'=_')
0082          READ(9,*)BUNCT(I,J)
0083          WRITE(9,71)J,I
0084    71    FORMAT(1X,'THE NUMBER OF SYSTEM SETUP HOURS PER DAY FOR ZONE',
0085         1I2,' OF UNIT',I2,'=_')
0086          READ(9,*)SHPDB(I,J)
0087          WRITE(9,72)J,I
0088    72    FORMAT(1X,'THE SYSTEM SETUP COST ($/HR) FOR ZONE',
0089         1I2,' OF UNIT',I2,'=_')
0090          READ(9,*)SCPHB(I,J)
0091          WRITE(9,20)
0092    20    FORMAT(1X,'HELISTAT TARGET LOAD IN POUNDS=@_')
0093          READ(9,*)TLOAD(I,J)
0094          PHPDB(I,J)=HPWDB(I,J)-SHPDB(I,J)
0095    16    CONTINUE
0096    C   CALCULATE ZONE DATA FOR EACH UNIT
0097    C   I=UNIT              J=ZONE IN UNIT I
0098    C   BNZN(IJ)=NO. OF BUNCHES IN ZONE J OF UNIT I
0099    C   TM(I,J)=TIME TO BUNCH ZONE J ON UNIT I
0100    C   PR(I,J)=PRODUCTION(TONS) FOR ZONE J  UNIT I
0101    C   CT(I,J)=COST TO BUNCH ZONE J   UNIT I
0102    C   PHPDB(I,J)=PRODUCTIVE HOURS PER DAY BUNCHING
0103    C   SHPDB(I,J)=SETUP HOURS PER DAY BUNCHING
0104    C   SCPHB(I,J)=SETUP COST PER HOUR BUNCHING
0105    C   HPWDB(I,J)=HOURS PER WORD DAY BUNCHING
0106          DO 66 I=1,UNITS
```

```
0107          N=ZUNIT(I)
0108          DO 14 J=1,N
0109          TM(I,J)=FELLNC(I,J)*BUNTM(I,J)/60.
0110          PR(I,J)=FELLNC(I,J)*FELLPR(I,J)/2000.
0111          CT(I,J)=TM(I,J)*BUNCT(I,J)
0112          BCNT(I,J)=PR(I,J)*2000./TLOAD(I,J)
0113          M=IFIX(BCNT(I,J)+0.99999)
0114          BNZN(I,J)=M
0115    C
0116    C     CALCULATE PER DAY DATA
0117    C
0118    C     TMDB(I,J)=TIME TO BUNCH IN DAYS
0119    C     SCDB(I,J)=SETUP COST IN DOLLARS
0120    C     TCFVB(I,J)=TOTAL COST FIXED AND VARIABLE
0121    C
0122          TMDB(I,J)=TM(I,J)/PHPDB(I,J)
0123          IC=TMDB(I,J)+1.0
0124          XIC=IC
0125          SCDB(I,J)=XIC*SHPDB(I,J)*SCPHB(I,J)
0126          TCFVB(I,J)=CT(I,J)+SCDB(I,J)
0127    C     CALCULATE $/TON   TONS/HR $/HR  ALL BY ZONE
0128    C     CPT(I,J)=COST PER TON FOR ZONE J UNIT I
0129    C     TPH(I,J)=TONS PER HOUR FOR ZONE J UNIT I
0130    C     CPH(I,J)=COST PER HOUR FOR ZONE J UNIT I
0131          CPT(I,J)=CT(I,J)/PR(I,J)
0132          TPH(I,J)=PR(I,J)/TM(I,J)
0133          CPH(I,J)=CT(I,J)/TM(I,J)
0134    C
0135    C     CALCULATE UNIT TOTALS
0136    C     BN(I)=NO. BUNCHES FOR UNIT I
0137    C     TTMU(I)=TIME FOR UNIT I
0138    C     PPRU(I)=PRODUCTION FOR UNIT I
0139    C     CCTU(I)=COST FOR UNIT I
0140          TTMU(I)=TM(I,J)+TTMU(I)
0141          PPRU(I)=PR(I,J)+PPRU(I)
0142          CCTU(I)=CT(I,J)+CCTU(I)
0143          BN(I)=BN(I)+BNZN(I,J)
0144    C
0145    C     CALCULATE UNIT TOTALS ON PER DAY BASIS
0146    C
0147    C     TDAUB(I)=TOTAL PRODUCTIVE TIME
0148    C     TFVUB(I)=TOTAL FIXED AND VARIABLE COST
0149    C     CSCDB(I)=TOTAL SETUP COST
0150    C
0151          TDAUB(I)=TDAUB(I)+TMDB(I,J)
0152          CSCDB(I)=CSCDB(I)+SCDB(I,J)
0153          TFVUB(I)=TFVUB(I)+TCFVB(I,J)
0154       14    CONTINUE
0155    C
0156    C     CALCULATE SITE TOTALS
0157    C     TOTTM(2)=TOTAL TIME FOR ACTIVITY 2
0158    C     TOTPR(2)=TOTAL PRODUCTION(TONS)FOR ACTIVITY 2
0159    C     TOTCT(2)=TOTAL COST FOR ACTIVITY 2
0160          TOTTM(2)=TTMU(I)+TOTTM(2)
```

```
0161          TOTPR(2)=TOTPR(2)+PPRU(I)
0162          TOTCT(2)=TOTCT(2)+CCTU(I)
0163    C  CALCULATE $/HOUR TONS/HOUR $/TON ALL ALL BY UNIT
0164    C  CPTU(I)=COST PER TON FOR UNIT I
0165    C  TPHU(I)=TONS PER HOUR FOR UNIT I
0166    C  CPHU(I)=COST PER HOUR FOR UNIT I
0167          CPTU(I)=CCTU(I)/PPRU(I)
0168          TPHU(I)=PPRU(I)/TTMU(I)
0169          CPHU(I)=CCTU(I)/TTMU(I)
0170     66   CONTINUE
0171    C  OUTPUT BUNCHING DATA BY ZONE & UNIT
0172     74   CALL NEWPG
0173          WRITE(9,73)
0174     73   FORMAT(49X,'ENTER NUMBER FOR OUTPUT DESIRED',
0175         1///,39X,'0=NONE',/,39X,'1=PRODUCTIVITY AND COST BASED ON',
0176         2' PER HOUR DATA',/,39X,'2=PRODUCTIVITY AND COST BASED ON',
0177         3' PER DAY DATA')
0178          READ(9,*)IK
0179          CALL NEWPG
0180          IF (IK) 74,78,75
0181     75   GO TO (76,77),IK
0182     77   NUNITS=UNITS
0183          DO 85 I=1,NUNITS
0184          WRITE(9,24)
0185     90   FORMAT(56X,'(PRODUCTIVE HOURS)')
0186          WRITE(9,17)I
0187          WRITE(9,80)
0188     80   FORMAT(36X,'TIME',34X,'COST (DOLLARS)',14X,'PRODUCTION (TONS)')
0189          WRITE(9,86)
0190     86   FORMAT(18X,38('*'),9X,30('*'),7X,17('*'))
0191          WRITE(9,81)
0192     81   FORMAT(18X,'PRODUCTIVE',9X,'SETUP',9X,'TOTAL',
0193         19X,'PRODUCTIVE',5X,'SETUP',5X,'TOTAL')
0194          WRITE(9,88)
0195     88   FORMAT(19X,'(HR/DAY)',7X,'(HR/DAY)',7X,'(DAYS)',//)
0196          N=ZUNIT(I)
0197          DO 83 J=1,N
0198          WRITE(9,82)J,PHPDB(I,J),SHPDB(I,J),TMDB(I,J),CT(I,J),
0199         1SCDB(I,J),TCFVB(I,J),PR(I,J)
0200     82   FORMAT(2X,'ZONE(',I2,')',7X,F8.2,9X,F8.2,5X,
0201         1F8.2,12X,F8.2,3X,F8.2,2X,F8.2,11X,F8.2)
0202     83   CONTINUE
0203          WRITE(9,84)TDAUB(I),CCTU(I),CSCDB(I),TFVUB(I),PPRU(I)
0204     84   FORMAT(3X,'TOTAL',39X,F8.2,12X,F8.2,3X,F8.2,2X,F8.2,11X,
0205         1F8.2,///)
0206     85   CONTINUE
0207          WRITE(9,31)
0208          CALL TINPT(ICHAR)
0209          GO TO 74
0210     76   NUNITS=UNITS
0211          DO 22 I=1,NUNITS
0212          WRITE(9,24)
0213     24   FORMAT(61X,'BUNCHING')
0214          WRITE(9,90)
```

```
0215          WRITE(9,17)I
0216          WRITE(9,18)
0217     18   FORMAT(22X,'HOURS',10X,'TONS',10X,'COST',10X,'$/TON',
0218         110X,'TONS/HR',10X,'$/HR',10X,'# BUNCHES',//)
0219          N=ZUNIT(I)
0220          DO 22 J=1,N
0221          WRITE(9,21)J,TM(I,J),PR(I,J),CT(I,J),CPT(I,J),TPH(I,J),CPH(I,J),
0222         1BNZN(I,J)
0223     21   FORMAT(4X,'ZONE(',I2,')',7X,F8.2,6X,F8.2,6X,F8.2,7X,
0224         2F8.2,8X,F8.2,7X,F8.2,7X,F8.2)
0225          IF(J.NE.N) GO TO 22
0226          WRITE(9,23)TTMU(I),PPRU(I),CCTU(I),CPTU(I),TPHU(I),CPHU(I),
0227         1BN(I)
0228     23   FORMAT(6X,'TOTALS',7X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,8X,
0229         3F8.2,7X,F8.2,7X,F8.2,////)
0230     22   CONTINUE
0231          WRITE(9,31)
0232     31   FORMAT(1X,'PRESS RETURN KEY TO CONTINUE')
0233          CALL TINPT(ICHAR)
0234          GO TO 74
0235    C    WRITE OUT UNIT INFO TO DATA FILE BNCH#
0236     78   CALL SPOLU(LU(2),BNCH1,1,1,25)
0237          DO 62 I=1,NUNITS
0238          WRITE(LU(2),61)TTMU(I),PPRU(I),CCTU(I)
0239     61   FORMAT(1X,10(F10.2,1X))
0240     62   CONTINUE
0241          DO 63 I=1,NUNITS
0242          N=ZUNIT(I)
0243          DO 64 J=1,N
0244          WRITE(LU(2),65)BUNTM(I,J),BUNCT(I,J),TLOAD(I,J),HPWDB(I,J),
0245         1PHPDB(I,J),SHPDB(I,J),SCPHB(I,J),TMDB(I,J)
0246     65   FORMAT(8(F8.2,1X))
0247     64   CONTINUE
0248     63   CONTINUE
0249     17   FORMAT(57X,'LOGGING UNIT(',I2,')',//)
0250    D     WRITE(6,800)LU(1),LU(2)
0251    D800  FORMAT(1X,'LU1,2=',2I8)
0252          CALL SPOLU(LU(2),BNCH1,1,2,25)
0253          CALL EXEC(14,2,COMMON,IBUFL)
0254          END
```

```
&YARD4 T=00004 IS ON CR00025 USING 00032 BLKS R=0041

0001  FTN4
0002        PROGRAM YARD
0003  C
0004  C COMPUTES THE TOTAL COST, TIME, & PRODUCTION OF ACTIVITY 3
0005        COMMON ITEK(45)
0006        INTEGER ZUNIT,UNITS
0007        COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
0008       1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
0009       2PHPDY(5,5)
0010        DIMENSION YARDCT(5,5),
0011       2TM(5,5),PR(5,5),CT(5,5),CPT(5,5),TPH(5,5),CPH(5,5),
0012       3TMU(5,5),PRU(5,5),CTU(5,5),CPTU(5),TPHU(5),CPHU(5)
0013        DIMENSION TTMU(5),PPRU(5),CCTU(5),BN(5),HPWDY(5,5),
0014       1TCFVY(5,5),SHPDY(5,5),SCPHY(5,5),TMDY(5,5),SCDY(5,5),
0015       2TFVUY(5),TDAUY(5),CSCDY(5)
0016        INTEGER COMMON(1),IPRM(5),YARD1(3)
0017        EQUIVALENCE (COMMON(1),ITEK(1))
0018        DATA YARD1 /2HYA, 2HRD, 2H# /
0019  C
0020        CALL RMPAR (IPRM)
0021  C
0022        IBUFL=390
0023        CALL EXEC (14,1,COMMON,IBUFL)
0024  D     WRITE(6,9)(COMMON(I),I=1,390)
0025  D9    FORMAT(1X,'INSIDE YARD COMMON IS',//,
0026  D    &49(10(2X,I8),/))
0027  C
0028        TOTCT(3)=0.
0029        TOTPR(3)=0.
0030        TOTTM(3)=0.
0031        NUNITS=UNITS
0032  C
0033        DO 27 I=1,UNITS
0034        N= ZUNIT(I)
0035        TTMU(I)=0.
0036        PPRU(I)=0.
0037        CCTU(I)=0.
0038        CPTU(I)=0.
0039        TPHU(I)=0.
0040        CPHU(I)=0.
0041        TDAUY(I)=0.
0042        CSCDY(I)=0.
0043        TFVUY(I)=0.
0044        BN(I)=0.
0045        DO 27 J=1,N
0046        HPWDY(I,J)=0.
0047        PHPDY(I,J)=0.
0048        SHPDY(I,J)=0.
0049        SCPHY(I,J)=0.
0050        TMDY(I,J)=0.
0051        SCDY(I,J)=0.
0052        TCFVY(I,J)=0.
```

```
0053          YARDTM(I,J)=0.
0054          YARDCT(I,J)=0.
0055          TM(I,J)=0.
0056          PR(I,J)=0.
0057          CT(I,J)=0.
0058          CPT(I,J)=0.
0059          TPH(I,J)=0.
0060          CPH(I,J)=0.
0061          TMU(I,J)=0.
0062          PRU(I,J)=0.
0063          CTU(I,J)=0.
0064     27   CONTINUE
0065          WRITE(9,50)
0066     50   FORMAT(1X,'INPUT YARDING DATA')
0067          DO 16 I=1,UNITS
0068          N=ZUNIT(I)
0069          DO 16 J=1,N
0070          WRITE(9,10)J,I
0071     10   FORMAT(///,1X,'YARDING CYCLE TIME(MIN)FOR ZONE',I2,' OF UNIT',I2,
0072         1'=_')
0073          READ(9,*)YARDTM(I,J)
0074          WRITE(9,70)J,I
0075     70   FORMAT(1X,'THE NUMBER OF HOURS PER WORK DAY',
0076         1' FOR ZONE',I2,' OF UNIT',I2,'=_')
0077          READ(9,*)HPWDY(I,J)
0078          WRITE(9,30)J,I
0079     30   FORMAT(1X,'SYSTEM YARDING COST/HR FOR ZONE',I2,' OF UNIT',I2,
0080         1'=_')
0081          READ(9,*)YARDCT(I,J)
0082          WRITE(9,71)J,I
0083     71   FORMAT(1X,'THE NUMBER OF SYSTEM SETUP HOURS PER DAY',
0084         1' FOR ZONE',I2,' OF UNIT',I2,'=_')
0085          READ(9,*)SHPDY(I,J)
0086          WRITE(9,72)J,I
0087     72   FORMAT(1X,'THE SYSTEM SETUP COST ($/HR) FOR ZONE',
0088         1I2,' OF UNIT',I2,'=_')
0089          READ(9,*)SCPHY(I,J)
0090          PHPDY(I,J)=HPWDY(I,J)-SHPDY(I,J)
0091     16   CONTINUE
0092     C    CALCULATE ZONE DATA FOR EACH UNIT
0093     C    I=UNIT              J=ZONE IN UNIT I
0094     C    TM(I,J)=TIME TO YARD  ZONE J ON UNIT I
0095     C    PR(I,J)=PRODUCTION(TONS) FOR UNIT J  UNIT I
0096     C    CT(I,J)=COST TO YARD  UNIT J   UNIT I
0097     C    HPWDY(I,J)=HOURS PER WORK DAY
0098     C    PHPDY(I,J)=PRODUCTIVE HOURS PER DAY
0099     C    SHPDY(I,J)=SETUP HOURS PER DAY
0100     C    SCPHY(I,J)=SETUP COST PER HOUR
0101          DO 66 I=1,UNITS
0102          N=ZUNIT(I)
0103          DO 14 J=1,N
0104          TM(I,J)=BNZN(I,J)*YARDTM(I,J)/60.
0105          PR(I,J)=(FELLPR(I,J)*FELLNC(I,J))/2000.
0106          CT(I,J)=TM(I,J)*YARDCT(I,J)
```

```
0107  C
0108  C  CALCULATE PER DAY DATA
0109  C
0110  C  TMDY(I,J)=TIME TO YARD A ZONE IN DAYS
0111  C  SCDY(I,J)=SETUP COST IN DOLLARS
0112  C  TCFVY(I,J)=TOTAL COST  FIXED AND VARIABLE
0113  C
0114         TMDY(I,J)=TM(I,J)/PHPDY(I,J)
0115         IC=TMDY(I,J)+1.0
0116         XIC=IC
0117         SCDY(I,J)=SCPHY(I,J)*XIC*SHPDY(I,J)
0118         TCFVY(I,J)=CT(I,J)+SCDY(I,J)
0119  C  CALCULATE $/TON   TONS/HR $/HR  ALL BY ZONE
0120  C  CPT(I,J)=COST PER TON FOR ZONE J UNIT I
0121  C  TPH(I,J)=TONS PER HOUR FOR ZONE J UNIT I
0122  C  CPH(I,J)=COST PER HOUR FOR ZONE J UNIT I
0123         CPT(I,J)=CT(I,J)/PR(I,J)
0124         TPH(I,J)=PR(I,J)/TM(I,J)
0125         CPH(I,J)=CT(I,J)/TM(I,J)
0126  C
0127  C  CALCULATE UNIT TOTALS
0128  C  TTMU(I)=TIME FOR UNIT I
0129  C  PPRU(I)=PRODUCTION FOR UNIT I
0130  C  CCTU(I)=COST FOR UNIT I
0131         TTMU(I)=TM(I,J)+TTMU(I)
0132         PPRU(I)=PR(I,J)+PPRU(I)
0133         CCTU(I)=CT(I,J)+CCTU(I)
0134         BN(I)=BN(I)+BNZN(I,J)
0135  C
0136  C  CALCULATE UNIT TOTALS ON PER DAY BASIS
0137  C
0138  C  TDAUY(I)=TOTAL PRODUCTIVE TIME TO YARD
0139  C  CSCDY(I)=TOTAL SETUP COST
0140  C  TFVUY(I)=TOTAL FIXED AND VARIALBLE COST
0141  C
0142         TDAUY(I)=TDAUY(I)+TMDY(I,J)
0143         CSCDY(I)=CSCDY(I)+SCDY(I,J)
0144         TFVUY(I)=TFVUY(I)+TCFVY(I,J)
0145     14  CONTINUE
0146  C
0147  C  CALCULATE SITE TOTALS
0148  C  TOTTM(3)=TOTAL TIME FOR ACTIVITY 3
0149  C  TOTPR(3)=TOTAL PRODUCTION(TONS)FOR ACTIVITY 3
0150  C  TOTCT(3)=TOTAL COST FOR ACTIVITY 3
0151         TOTTM(3)=TTMU(I)+TOTTM(3)
0152         TOTPR(3)=TOTPR(3)+PPRU(I)
0153         TOTCT(3)=TOTCT(3)+CCTU(I)
0154  C  CALCULATE $/HOUR TONS/HOUR $/TON ALL ALL BY UNIT
0155  C  CPTU(I)=COST PER TON FOR UNIT I
0156  C  TPHU(I)=TONS PER HOUR FOR UNIT I
0157  C  CPHU(I)=COST PER HOUR FOR UNIT I
0158         CPTU(I)=CCTU(I)/PPRU(I)
0159         TPHU(I)=PPRU(I)/TTMU(I)
0160         CPHU(I)=CCTU(I)/TTMU(I)
```

```
0161   66   CONTINUE
0162  C  OUTPUT YARDING DATA BY ZONE & UNIT
0163    74  CALL NEWPG
0164        WRITE(9,73)
0165   73   FORMAT(49X,'ENTER NUMBER FOR OUTPUT DESIRED',
0166       1///,39X,'0=NONE',/,39X,'1=PRODUCTIVITY AND COST BASED ON',
0167       2' PER HOUR DATA',/,39X,'2=PRODUCTIVITY AND COST BASED ON',
0168       3' PER DAY DATA')
0169        READ(9,*)IK
0170        CALL NEWPG
0171        IF(IK)74,78,75
0172   75   GO TO (76,77),IK
0173   77   NUNITS=UNITS
0174        DO 85 I=1,NUNITS
0175   90   FORMAT(56X,'(PRODUCTIVE HOURS)')
0176        WRITE(9,24)
0177        WRITE(9,17)I
0178        WRITE(9,80)
0179   80   FORMAT(36X,'TIME',34X,'COST (DOLLARS)',14X,'PRODUCTION (TONS)')
0180        WRITE(9,86)
0181   86   FORMAT(18X,38('*'),9X,30('*'),7X,17('*'))
0182        WRITE(9,81)
0183   81   FORMAT(18X,'PRODUCTIVE',9X,'SETUP',9X,'TOTAL',
0184       19X,'PRODUCTIVE',5X,'SETUP',5X,'TOTAL')
0185        WRITE(9,88)
0186   88   FORMAT(19X,'(HR/DAY)',7X,'(HR/DAY)',7X,'(DAYS)',//)
0187        N=ZUNIT(I)
0188        DO 83 J=1,N
0189        WRITE(9,82)J,PHPDY(I,J),SHPDY(I,J),TMDY(I,J),
0190       1CT(I,J),SCDY(I,J),TCPVY(I,J),PR(I,J)
0191   82   FORMAT(2X,'ZONE(',I2,')',7X,F8.2,9X,F8.2,5X,
0192       1F8.2,12X,F8.2,3X,F8.2,3X,F8.2,11X,F8.2)
0193   83   CONTINUE
0194        WRITE(9,84)TDAUY(I),CCTU(I),CSCDY(I),TPVUY(I),PPRU(I)
0195   84   FORMAT(2X,'TOTAL',40X,F8.2,12X,F8.2,3X,F8.2,3X,
0196       1F8.2,11X,F8.2,///)
0197   85   CONTINUE
0198        WRITE(9,31)
0199        CALL TINPT(ICHAR)
0200        GO TO 74
0201   76   NUNITS=UNITS
0202        DO 22 I=1,NUNITS
0203        WRITE(9,24)
0204        WRITE(9,90)
0205   24   FORMAT(61X,'YARDING')
0206        WRITE(9,17)I
0207        WRITE(9,18)
0208   18   FORMAT(22X,'HOURS',10X,'TONS',10X,'COST',10X,'$/TON',
0209       110X,'TONS/HR',10X,'$/HR',10X,'# BUNCHES',//)
0210        N=ZUNIT(I)
0211        DO 22 J=1,N
0212        WRITE(9,21)J,TM(I,J),PR(I,J),CT(I,J),CPT(I,J),TPH(I,J),CPH(I,J),
0213       1BNZN(I,J)
0214   21   FORMAT(4X,'ZONE(',I2,')',7X,F8.2,6X,F8.2,6X,F8.2,7X,
```

```
0215        2F8.2,8X,F8.2,7X,F8.2,7X,F8.2)
0216         IF(J.NE.N) GO TO 22
0217         WRITE(9,23)TTMU(I),PPRU(I),CCTU(I),CPTU(I),TPHU(I),CPHU(I),BN(I)
0218   23    FORMAT(6X,'TOTALS',7X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,8X,
0219        3F8.2,7X,F8.2,7X,F8.2,////)
0220   22    CONTINUE
0221         WRITE(9,31)
0222   31    FORMAT(1X,'PRESS RETURN KEY TO CONTINUE')
0223         CALL TINPT(ICHAR)
0224         GO TO 74
0225   17    FORMAT(57X,'LOGGING UNIT(',I2,')',//)
0226  C  WRITE OUT UNIT INFO TO DATA FILE (#YARD)
0227   78    CALL SPOLU (LU(3),YARD1,1,1,25)
0228         DO 62 I=1,NUNITS
0229         WRITE(LU(3),61) TTMU(I),PPRU(I),CCTU(I)
0230   61    FORMAT(1X,10(F10.2,1X))
0231   62    CONTINUE
0232         DO 63 I=1,NUNITS
0233         N=ZUNIT(I)
0234         DO 64 J=1,N
0235         WRITE(LU(3),65)YARDTM(I,J),YARDCT(I,J),HPWDY(I,J),
0236        1PHPDY(I,J),SHPDY(I,J),SCPHY(I,J),TMDY(I,J)
0237   65    FORMAT(7(F8.2,1X))
0238   64    CONTINUE
0239   63    CONTINUE
0240  D      WRITE(6,800)LU(1),LU(2),LU(3)
0241  D800   FORMAT(1X,'LU1,2,3',3I8)
0242         CALL SPOLU(LU(3),YARD1,1,2,25)
0243         CALL EXEC(14,2,COMMON,IBUFL)
0244         END
```

```
__AND4 T=00004 IS ON CR00025 USING 00030 BLKS R=0000

 001   FTN4
 002         PROGRAM LAND
0003   C
^004   C COMPUTES THE TOTAL COST, TIME, & PRODUCTION OF ACTIVITY 4
 005         COMMON ITEK(45)
0006         INTEGER ZUNIT,UNITS
0007         COMMON UNITS,ZUNIT(5),ISTA(5),TOTTM(5),TOTPR(5),TOTCT(5),LU(4),
 008        1FELLNC(5,5),FELLPR(5,5),TLOAD(5,5),BNZN(5,5),YARDTM(5,5),
 009        2PHPDY(5,5)
0010         DIMENSION LANDTM(5,5),LANDNC(5,5),LANDCT(5,5),LANDPR(5,5),
0011        2TM(5,5),PR(5,5),CT(5,5),CPT(5,5),TPH(5,5),CPH(5,5),
0012        3TMU(5,5),PRU(5,5),CTU(5,5),CPTU(5),TPHU(5),CPHU(5)
0013         DIMENSION TTMU(5),PPRU(5),CCTU(5),BN(5),PHPDL(5,5),
0014        1SHPDL(5,5),SCPHL(5,5),HPWDL(5,5),TCFVL(5,5),TMDL(5,5),
0015        2SCDL(5,5),TDAUL(5),CSCDL(5),TFVUL(5)
0016         INTEGER COMMON(1),IPRM(5),LAND1(3)
0017         EQUIVALENCE (COMMON(1),ITEK(1))
0018         DATA LAND1/ 2HLA,2HND,2H# /
0019   C
0020         CALL RMPAR (IPRM)
0021   C
0022         IBUFL=390
0023         CALL EXEC (14,1,COMMON,IBUFL)
0024   D     WRITE(6,9)(COMMON(I),I=1,390)
0025   D9    FORMAT(1X,'INSIDE LAND COMMON IS',//,
0026   D    &49(10(2X,I8),/))
0027   C
0028         TOTCT(4)=0.
0029         TOTPR(4)=0.
0030         TOTTM(4)=0.
0031         NUNITS=UNITS
0032   C
0033         DO 27 I=1,UNITS
0034         N= ZUNIT(I)
0035         TTMU(I)=0.
0036         PPRU(I)=0.
0037         CCTU(I)=0.
0038         CPTU(I)=0.
0039         TPHU(I)=0.
0040         TDAUL(I)=0.
0041         CSCDL(I)=0.
0042         TFVUL(I)=0.
0043         CPHU(I)=0.
0044         BN(I)=0.
0045         DO 27 J=1,N
0046         LANDTM(I,J)=0.
0047         LANDNC(I,J)=0.
0048         LANDCT(I,J)=0.
0049         LANDPR(I,J)=0.
0050         TM(I,J)=0.
0051         PR(I,J)=0.
0052         CT(I,J)=0.
```

```
0053          CPT(I,J)=0.
0054          TPH(I,J)=0.
0055          CPH(I,J)=0.
0056          TMU(I,J)=0.
0057          PRU(I,J)=0.
0058          CTU(I,J)=0.
0059          TMDL(I,J)=0.
0060          SCDL(I,J)=0.
0061          TCFVL(I,J)=0.
0062          PHPDL(I,J)=0.
0063          SHPDL(I,J)=0.
0064          SCPHL(I,J)=0.
0065    27    CONTINUE
0066          WRITE(9,50)
0067    50    FORMAT(1X,'INPUT THE LANDING DATA')
0068          DO 16 I=1,UNITS
0069          N=ZUNIT(I)
0070          DO 16 J=1,N
0071          WRITE(9,30)J,I
0072    30    FORMAT(///,1X,'SYSTEM LANDING COST/HR FOR ZONE',I2,
0073         1' OF UNIT',I2,'=_')
0074          READ(9,*)LANDCT(I,J)
0075          PHPDL(I,J)=PHPDY(I,J)
0076          WRITE(9,70)J,I
0077    70    FORMAT(1X,'THE NUMBER OF SYSTEM SETUP HOURS PER DAY',
0078         1' FOR ZONE',I2,' OF UNIT',I2,'=_')
0079          READ(9,*)SHPDL(I,J)
0080          HPWDL(I,J)=PHPDL(I,J)+SHPDL(I,J)
0081          WRITE(9,71)J,I
0082    71    FORMAT(1X,'THE SYSTEM SETUP COST PER HOUR FOR ZONE',I2,
0083         1' OF UNIT',I2,'=_')
0084          READ(9,*)SCPHL(I,J)
0085    16    CONTINUE
0086    C  CALCULATE ZONE DATA FOR EACH UNIT
0087    C  I=UNIT              J=ZONE IN UNIT I
0088    C  TM(I,J)=TIME TO LAND  ZONE J ON UNIT I
0089    C  PR(I,J)=PRODUCTION(TONS) FOR UNIT J  UNIT I
0090    C  CT(I,J)=COST TO LAND   UNIT J   UNIT I
0091          DO 66 I=1,UNITS
0092          N=ZUNIT(I)
0093          DO 14 J=1,N
0094          TM(I,J)=BNZN(I,J)*YARDTM(I,J)/60.
0095          PR(I,J)=(FELLPR(I,J)*FELLNC(I,J))/2000.
0096          CT(I,J)=TM(I,J)*LANDCT(I,J)
0097    C
0098    C  CALCULATE PER DAY DATA
0099    C
0100    C  TMDL(I,J)=TIME TO PROCESS YARDED LOGS
0101    C  SCDL(I,J)=SETUP COST IN DOLLARS
0102    C  TCFVL(I,J)=TOTAL COST  FIXED AND VARIABLE
0103    C
0104          TMDL(I,J)=TM(I,J)/PHPDL(I,J)
0105          IC=TMDL(I,J)+1.0
0106          XIC=IC
```

```
0107          SCDL(I,J)=SCPHL(I,J)*XIC*SHPDL(I,J)
0108          TCFVL(I,J)=CT(I,J)+SCDL(I,J)
0109    C   CALCULATE $/TON   TONS/HR $/HR  ALL BY ZONE
0110    C   CPT(I,J)=COST PER TON FOR ZONE J UNIT I
0111    C   TPH(I,J)=TONS PER HOUR FOR ZONE J UNIT I
0112    C   CPH(I,J)=COST PER HOUR FOR ZONE J UNIT I
0113          CPT(I,J)=CT(I,J)/PR(I,J)
0114          TPH(I,J)=PR(I,J)/TM(I,J)
0115          CPH(I,J)=CT(I,J)/TM(I,J)
0116    C
0117    C   CALCULATE UNIT TOTALS
0118    C   TTMU(I)=TIME FOR UNIT I
0119    C   PPRU(I)=PRODUCTION FOR UNIT I
0120    C   CCTU(I)=COST FOR UNIT I
0121          TTMU(I)=TM(I,J)+TTMU(I)
0122          PPRU(I)=PR(I,J)+PPRU(I)
0123          CCTU(I)=CT(I,J)+CCTU(I)
0124          BN(I)=BN(I)+BNZN(I,J)
0125    C
0126    C   CALCULATE ON PER DAY BASIS
0127    C
0128    C   TDAUL(I)=TOTAL PRODUCTIVE TIME
0129    C   CSCDL(I)=TOTAL SETUP COST
0130    C   TFVUL(I)=TOTAL COST
0131    C
0132          TDAUL(I)=TDAUL(I)+TMDL(I,J)
0133          CSCDL(I)=CSCDL(I)+SCDL(I,J)
0134          TFVUL(I)=TFVUL(I)+TCFVL(I,J)
0135       14    CONTINUE
0136    C
0137    C   CALCULATE SITE TOTALS
0138    C   TOTTM(4)=TOTAL TIME FOR ACTIVITY 4
0139    C   TOTPR(4)=TOTAL PRODUCTION(TONS)FOR ACTIVITY 4
0140    C   TOTCT(4)=TOTAL COST FOR ACTIVITY 4
0141          TOTTM(4)=TTMU(I)+TOTTM(4)
0142          TOTPR(4)=TOTPR(4)+PPRU(I)
0143          TOTCT(4)=TOTCT(4)+CCTU(I)
0144    C   CALCULATE $/HOUR TONS/HOUR $/TON ALL ALL BY UNIT
0145    C   CPTU(I)=COST PER TON FOR UNIT I
0146    C   TPHU(I)=TONS PER HOUR FOR UNIT I
0147    C   CPHU(I)=COST PER HOUR FOR UNIT I
0148          CPTU(I)=CCTU(I)/PPRU(I)
0149          TPHU(I)=PPRU(I)/TTMU(I)
0150          CPHU(I)=CCTU(I)/TTMU(I)
0151       66    CONTINUE
0152    C   OUTPUT LANDING DATA BY ZONE & UNIT
0153       74    CALL NEWPG
0154          WRITE(9,73)
0155       73    FORMAT(49X,'ENTER NUMBER FOR OUTPUT DESIRED',
0156         1///,39X,'0=NONE',/,39X,'1=PRODUCTIVITY AND COST BASED ON',
0157         2' PER HOUR DATA',/,39X,'2=PRODUCTIVITY AND COST BASED ON',
0158         3' PER DAY DATA')
0159          READ(9,*)IK
0160          CALL NEWPG
```

```
0161          IF(IK) 74,78,75
0162     75   GO TO (76,77),IK
0163     77   NUNITS=UNITS
0164          DO 85 I=1,NUNITS
0165     90   FORMAT(56X,'(PRODUCTIVE HOURS)')
0166          WRITE(9,24)
0167          WRITE(9,17)I
0168          WRITE(9,80)
0169     80   FORMAT(36X,'TIME',34X,'COST (DOLLARS)',14X,'PRODUCTION (TONS)')
0170          WRITE(9,86)
0171     86   FORMAT(18X,38('*'),9X,30('*'),7X,17('*'))
0172          WRITE(9,81)
0173     81   FORMAT(18X,'PRODUCTIVE',9X,'SETUP',9X,'TOTAL',
0174         19X,'PRODUCTIVE',5X,'SETUP',5X,'TOTAL')
0175          WRITE(9,88)
0176     88   FORMAT(19X,'(HR/DAY)',7X,'(HR/DAY)',7X,'(DAYS)',//)
0177          N=ZUNIT(I)
0178          DO 83 J=1,N
0179          WRITE(9,82)J,PHPDL(I,J),SHPDL(I,J),TMDL(I,J),
0180         1CT(I,J),SCDL(I,J),TCFVL(I,J),PR(I,J)
0181     82   FORMAT(2X,'ZONE(',I2,')',7X,F8.2,9X,F8.2,5X,
0182         1F8.2,12X,F8.2,3X,F8.2,3X,F8.2,11X,F8.2)
0183     83   CONTINUE
0184          WRITE(9,84)TDAUL(I),CCTU(I),CSCDL(I),
0185         1TFVUL(I),PPRU(I)
0186     84   FORMAT(2X,'TOTAL',40X,F8.2,12X,F8.2,3X,F8.2,
0187         13X,F8.2,11X,F8.2,///)
0188     85   CONTINUE
0189          WRITE(9,31)
0190          CALL TINPT(ICHAR)
0191          GO TO 74
0192     76   NUNITS=UNITS
0193          DO 22 I=1,NUNITS
0194          WRITE(9,24)
0195     24   FORMAT(61X,'LANDING')
0196          WRITE(9,90)
0197          WRITE(9,17)I
0198          WRITE(9,18)
0199     18   FORMAT(22X,'HOURS',10X,'TONS',10X,'COST',10X,'$/TON',
0200         110X,'TONS/HR',10X,'$/HR',10X,'# BUNCHES',//)
0201          N=ZUNIT(I)
0202          DO 22 J=1,N
0203          WRITE(9,21)J,TM(I,J),PR(I,J),CT(I,J),CPT(I,J),TPH(I,J),CPH(I,J) ,
0204         1BNZN(I,J)
0205     21   FORMAT(4X,'ZONE(',I2,')',7X,F8.2,6X,F8.2,6X,F8.2,7X,
0206         2F8.2,8X,F8.2,7X,F8.2,7X,F8.2)
0207          IF(J.NE.N) GO TO 22
0208          WRITE(9,23)TTMU(I),PPRU(I),CCTU(I),CPTU(I),TPHU(I),CPHU(I),BN(I)
0209     23   FORMAT(6X,'TOTALS',7X,F8.2,6X,F8.2,6X,F8.2,7X,F8.2,8X,
0210         3F8.2,7X,F8.2,7X,F8.2,/////)
0211     22   CONTINUE
0212          WRITE(9,31)
0213     31   FORMAT(1X,'PRESS RETURN TO CONTINUE')
0214          CALL TINPT(ICHAR)
```

```
0215          GO TO 74
0216    17    FORMAT(57X,'LOGGING UNIT(',I2,')',//)
0217    C  WRITE OUT INFO TO DATA FILE LAND#
0218    78    CALL SPOLU(LU(4),LAND1,1,1,25)
0219          DO 62 I=1,NUNITS
0220          WRITE(LU(4),61)TTMU(I),PPRU(I),CCTU(I)
0221    61    FORMAT(1X,10(F10.2,1X))
0222    62    CONTINUE
0223          DO 63 I=1,NUNITS
0224          N=ZUNIT(I)
0225          DO 64 J=1,N
0226          WRITE(LU(4),65)LANDCT(I,J),SHPDL(I,J),SCPHL(I,J),HPWDL(I,J)
0227         1,PHPDL(I,J),TMDL(I,J)
0228    65    FORMAT(6(F8.2,1X))
0229    64    CONTINUE
0230    63    CONTINUE
0231    D     WRITE(6,800)LU(1),LU(2),LU(3),LU(4)
0232    D800  FORMAT(1X,'LU1,2,3,4',4I8)
0233          CALL SPOLU(LU(4),LAND1,1,2,25)
0234          CALL EXEC(14,2,COMMON,IBUFL)
0235          END
```

AD-33 Bookplate
(1-63)

NATIONAL
AGRICULTURAL
LIBRARY

LIBRARY FILE COPY
INTERMOUNTAIN STATION
OGDEN, UT.

ACCO

INTERMOUNTAIN FOREST
AND RANGE EXP. STATION
507 25th Street
Ogden, Utah 84401